



TUGAS AKHIR - KI141502

Desain dan Implementasi Rendering Sketsa pada Obyek Tiga Dimensi: Modul Pembangkitan Pena Dinamis

**IGNATIUS ABRAHAM SUSANTO
NRP 5112100048**

**Dosen Pembimbing
Wijayanti Nurul K., S.Kom., M.Sc.
Rully Soelaiman, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016**

(Halaman Ini Sengaja dikosongkan)



FINAL PROJECT- KI141502

Design and Implementation Sketch Rendering in 3D Object: Dynamic Pen Generation Module

**IGNATIUS ABRAHAM SUSANTO
NRP 5112100048**

**Dosen Pembimbing
Wijayanti Nurul K., S.Kom., M.Sc.
Rully Soelaiman, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2016**

(Halaman Ini Sengaja dikosongkan)

LEMBAR PENGESAHAN

Desain dan Implementasi Rendering Sketsa pada Obyek Tiga Dimensi: Modul Pembangunan Pena Dinamis Tugas Akhir

**Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Interaksi, Grafika, dan Seni
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember**

Oleh:
IGNATIUS ABRAHAM SUSANTO
NRP. 5112 100 048

Disetujui oleh Dosen Pembimbing tugas akhir

Wijayanti Nurul K., S.Kom., M.Sc.

NIP: 19860312 201212 2 004

(pembimbing 1)

Rully Soelaiman, S.Kom., M.Kom.

NIP: 19700213 199402 1 001

(pembimbing 2)



**SURABAYA
JUNI, 2016**

(Halaman Ini Sengaja dikosongkan)

Desain dan Implementasi Rendering Sketsa pada Obyek Tiga Dimensi: Modul Pembangkitan Pena Dinamis

Nama Mahasiswa : Ignatius Abraham Susanto
NRP : 5112 100 048
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Wijayanti Nurul K., S.Kom., M.Sc.
Dosen Pembimbing II : Rully Soelaiman, S.Kom., M.Kom.

ABSTRAK

Sketsa adalah salah satu karya seni untuk meniru atau menggambar secara kasar dari sudut pandang tertentu. Untuk membuat sketsa pada umumnya memerlukan suatu alat. Alat tersebut adalah pena sketsa yang dipengaruhi oleh gerakan tangan. Hal yang dapat dipengaruhi dari gerakan tangan adalah arah gerakan pena, penekanan tangan, ketebalan pena, dan cara menggariskan. Dalam ilmu komputer, hal ini dapat didekati dengan muscle model dan pergerakan pena.

Metode yang dilakukan pada tugas akhir ini adalah memodelkan otot dengan menggunakan PID Control dan mensimulasikan pergerakan pena dengan ode solver. PID Control ini digunakan untuk mengontrol trajektori referensi 3D, dan solver digunakan untuk pergerakan penanya setelah titik tersebut dikontrol.

Dari hasil uji coba untuk parameter PID dari tiga obyek berbeda, didapatkan bahwa parameter PID yang telah di-tuning memiliki performa sistem yang lebih stabil. Sedangkan dari hasil uji coba waktu eksekusi, solver yang memiliki performa yang lebih baik adalah ode23, ode113. Jadi untuk memodelkan otot, PID yang lebih baik adalah PID yang telah di-tuning dan untuk performa waktu yang lebih baik, pada sistem ini, ode23 dan ode113.

Kata kunci: Sketsa, pena sketsa, PID, ODE solver, otot tangan, pergerakan pena.

(Halaman Ini Sengaja dikosongkan)

Design and Implementation Sketch Rendering in 3D Object: Dynamic Pen Generation Module

Student Name : Ignatius Abraham Susanto
NRP : 5112 100 048
Major : Teknik Informatika FTIf-ITS
Advisor I : Wijayanti Nurul K., S.Kom., M.Sc.
Advisor II : Rully Soelaiman, S.Kom., M.Kom.

ABSTRACT

Sketching is one of the works of art to imitate or rough drawing of a particular viewpoint. To sketch usually require a tool. The tool is a pen sketch that is influenced by the movement of the hand. Things can be influenced from hand movement is the direction of pen movement, suppression hand, the thickness of the pen, and outline ways. In computer science, this can be approximated by a muscle model and movement of the pen.

The method used in this final project is modeling the muscles by using PID Control and simulate the movement of the pen with the ode solver. PID Control is used to control the trajectory of reference 3D, and solver used for the movement of his pen after that point controlled.

From the test results for the PID parameters of three different objects, it was found that the PID parameters that have been tuning have a more stable system performance. While the results of the test execution time, solver who has the better performance is ode23, ode113. So in order to model the muscles, the better PID is a PID that has been on-tuning and for a better performance, in this system, ode23 and ode113.

Keywords: Sketch, pen sketches, PID, ODE solver, the hand muscles, the movement of the pen

(Halaman Ini Sengaja dikosongkan)

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Metodologi	2
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 MATLAB, SIMULINK dan Solver.....	5
2.2 PID CONTROLLER	9
BAB III ANALISIS DAN PERANCANGAN.....	13
3.1 Desain Metode Secara Umum	13
3.2 Perancangan untuk Membaca Input	13
3.2.1 Membaca dari text file	16
3.2.2 Menambahkan Block Constant.....	18
3.3 Perancangan PID	19
3.3.1 Menambahkan Block PID.....	20
3.3.2 Setting Parameter PID	20
3.3.3 Penambahan <i>Block Sum</i>	21
3.4 Perancangan Pra-Simulasi	22
3.4.1 Menambahkan MATLAB Function Block	23
3.4.2 Menghubungkan Block.....	24
3.5 Perancangan Simulasi.....	24
3.5.1 Memilih <i>Solver</i>	25
3.5.2 Menjalankan Simulasi	25

BAB IV IMPLEMENTASI	27
4.1 Lingkungan Implementasi	27
4.2 Implementasi	27
4.2.1 Implementasi Membaca Input.....	27
4.2.2 Implementasi PID	30
4.2.3 Implementasi Pra-Simulasi	32
4.2.4 Implementasi Simulasi.....	37
BAB V PENGUJIAN DAN EVALUASI	43
5.1 Lingkungan Uji Coba	43
5.2 Data Uji Coba	43
5.3 Uji Coba Parameter PID	44
5.3.1 Skenario Uji Coba Parameter PID	45
5.3.2 Hasil Pengujian dan Evaluasi Parameter PID Skenario I	45
5.3.3 Hasil Pengujian dan Evaluasi Parameter PID Skenario II	45
5.4 Uji Coba Waktu Eksekusi Perangkat Lunak	49
5.4.1 Uji Coba dan Evaluasi Waktu Eksekusi Perangkat Lunak	49
BAB VI KESIMPULAN DAN SARAN.....	53
6.1. Kesimpulan.....	53
6.2. Saran	53
DAFTAR PUSTAKA.....	55
LAMPIRAN A. Referensi Trajektori 3D	57
LAMPIRAN B. MATLAB Function.....	79
LAMPIRAN C. Implementasi Sistem	85
BIODATA PENULIS.....	89

DAFTAR GAMBAR

Gambar 2.1 Pemilihan Solver pada SIMULINK	6
Gambar 2.2 Contoh Model Stiff : Foucault Pendulum Model [6].	7
Gambar 2.3 Struktur PID Controller	10
Gambar 3.1 Gambar Alur Proses Secara Umum	14
Gambar 3.2 Gambar Alur Proses Membaca Input	15
Gambar 3.3 Contoh Tampilan dari <i>Text File</i>	17
Gambar 3.4 <i>Pseudocode</i> Membaca dari Text File (Bagian Pertama)	17
.....	18
Gambar 3.5 <i>Pseudocode</i> Membaca dari Text File (Bagian Kedua)	18
.....	18
Gambar 3.6 <i>Pseudocode</i> Penambahan Block Constant.....	18
Gambar 3.7 Gambar Alur Proses PID	19
Gambar 3.8 <i>Pseudocode</i> Penambahan Block PID	20
Gambar 3.9 <i>Pseudocode Setting</i> Parameter PID	21
Gambar 3.10 <i>Pseudocode</i> Penambahan <i>Block Sum</i>	21
Gambar 3.11 Gambar Alur Proses Pra-Simulasi	22
Gambar 3.12 <i>Pseudocode</i> Penambahan Block Sum.....	24
Gambar 3.13 Gambar Alur Proses Simulasi.....	25
Gambar 4.1 Hasil dari Proses Membaca Input.....	28
Gambar 4.2 Hasil dari Proses PID.....	30
Gambar 4.3 Hasil dari Proses Pra-Simulasi	32
Gambar 4.4 Membuka SIMULINK	33
Gambar 4.5 Membuka Library Browser	33
Gambar 4.6 Memilih MATLAB Function	34
Gambar 4.7 Men- <i>drag</i> MATLAB Function pada lembar kerja	
SIMULINK	34
Gambar 4.8 Mengubah isi MATLAB Function	35
Gambar 4.9 Menambah block constant dummy	37
Gambar 4.10 Hasil Dari Proses Simulasi Dengan Input <i>Segitiga.txt</i>	38
.....	38
Gambar 4.11 Hasil Dari Proses Simulasi Dengan Input <i>Cube.txt</i>	38
Gambar 4.12 Hasil Dari Proses Simulasi Dengan Input	
<i>Segilima.txt</i>	39

Gambar 4.13 Membuka SIMULINK.....39

Gambar 4.14 Memilih Model Configuration Parameter.....40

Gambar 4.15 Memilih solver.....40

Gambar 5.1 Objek 3D Untuk Diuji44

Gambar 5.2 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Kubus.....46

Gambar 5.3 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Segilima.....46

Gambar 5.4 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Segitiga.....47

Gambar 5.5 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Kubus.....47

Gambar 5.6 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Segilima.....48

Gambar 5.7 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Segitiga.....48

DAFTAR TABEL

Tabel 2.1 Karakteristik ode45, ode23, ode113 (Bagian Pertama) ..	8
Tabel 2.2 Karakteristik ode45, ode23, ode113 (Bagian Kedua) ...	9
Tabel 2.3 Karakteristik PID.....	11
Tabel 2.4 Tabel Tuning Ziegler-Nichols.....	11
Tabel 3.1 Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Membaca Input (Bagian Pertama).....	15
Tabel 3.2 Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Membaca Input (Bagian Kedua)	16
Tabel 3.3 Daftar Fungsi yang Digunakan Pada <i>Pseudocode</i> Membaca Input.....	16
Tabel 3.4 Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> PID	19
Tabel 3.5 Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Pra-Simulasi (Bagian Pertama).....	22
Tabel 3.6 Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Pra-Simulasi (Bagian Kedua).....	23
Tabel 5.1 Hasil Uji Coba Waktu Eksekusi Objek Cube.....	49
Tabel 5.2 Hasil Uji Coba Waktu Eksekusi Objek Segilima	50
Tabel 5.3 Hasil Uji Coba Waktu Eksekusi Objek Segitiga	50
Tabel 5.4 Rata-rata Hasil Uji Coba Waktu Eksekusi (Bagian Pertama)	51
Tabel 5.5 Rata-rata Hasil Uji Coba Waktu Eksekusi (Bagian Kedua)	52

(Halaman Ini Sengaja dikosongkan)

BAB I

PENDAHULUAN

Bab ini memaparkan garis besar tugas akhir yang meliputi latar belakang, tujuan dan manfaat pembuatan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1 Latar Belakang

Sketsa adalah salah satu karya seni untuk meniru atau menggambar secara kasar dari sudut pandang tertentu. Umumnya sketsa digunakan untuk membantu mengingat objek yang digambar. Jadi sketsa adalah gambar kasar dan bersifat sementara dengan memiliki nilai seni dari suatu objek yang digambar.

Untuk membuat sketsa pada umumnya memerlukan alat yang sangat sederhana untuk dapat membuat tanda goresan yang mewakili bentuk sesungguhnya. Alat itu tidak lain adalah pena sketsa yang dipengaruhi oleh gerakan tangan. Hal yang dapat dipengaruhi dari gerakan tangan adalah arah gerakan pena, penekanan tangan, ketebalan pena, dan cara menggariskan.

Sketsa yang merupakan hasil karya dari manusia, mulai dicoba diterapkan pada komputer dari pola pergerakan tangan manusia. Sehingga seolah-olah komputer dapat membuat penggarisan secara dinamis [1]. Penggarisan yang dimaksud adalah *silhouette* dari objek 3D. Penelitian ini [1] menginspirasi perkembangan yang lebih lanjut, yang mana diterapkan dalam tugas akhir ini dalam pena dinamis.

Pada tugas akhir ini akan dilakukan pembuatan pena dinamis dalam render obyek 3D untuk membuat penggambaran sketsa. Pena dinamis yang dibuat memodelkan proses pergerakan otot tangan manusia [2]. Pena dinamis adalah proses pergerakan pena yang memiliki input 3D referensi trajektori dengan melibatkan aspek *dynamic modelling* sebagai proses utamanya. Aspek *dynamic modelling* yang digunakan adalah bagaimana memodelkan pergerakan tangan manusia dan pergerakan pena.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Memahami konsep dasar dari pemodelan pergerakan tangan manusia untuk dijadikan model sebagai pergerakan pena dinamis
2. Bagaimana mengimplementasikan pena dinamis untuk rendering 3D objek dalam bentuk sketsa pensil?
3. Bagaimana melakukan pengujian agar hasil dapat diterima dengan baik?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Aplikasi yang dibangun dengan menggunakan SIMULINK versi 8.1 dan MATLAB 2013a
2. Batasan input port pada MATLAB Function adalah 34.
3. Data referensi trajektori yang dipakai adalah Cube.txt, Segilima.txt, Segitiga.txt.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Membuat aplikasi pena dinamis untuk penelusuran dan penggambaran sketsa di dalam objek 3D.
2. Menguji kinerja aplikasi yang meliputi aspek kestabilan sistem dan kinerja waktu.

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini adalah mempermudah proses mensimulasikan pergerakan pena secara dinamis.

1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

A. Studi literatur

Tahap Studi Literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan tugas akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi dan pustaka, serta pemahaman dasar teori yang digunakan pada topik tugas akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. Penggunaan MATLAB;
2. Penggunaan SIMULINK;
3. Penerapan PID;
4. Penerapan Solver;

B. Perancangan perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

C. Implementasi dan pembuatan sistem

Tahap implementasi merupakan tahap untuk membangun aplikasi permainan beserta sistem yang terkait. Aplikasi ini akan dibangun dengan bahasa pemrograman MATLAB.

D. Uji coba dan evaluasi

Pada tahapan ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan juga untuk mengevaluasi jalannya sistem, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

E. Penyusunan laporan tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

1.7 Sistematika Penulisan

Buku tugas akhir ini terdiri dari beberapa bab, yang dijelaskan sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan tugas akhir, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir ini.

BAB III ANALISIS DAN PERANCANGAN

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *pseudocode*.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembangkitan muscle model, dan pen motion.

BAB V PENGUJIAN DAN EVALUASI

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian pengujian kinerja dari sistem yang telah dibuat.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan tugas akhir. Teori-teori tersebut adalah MATLAB, SIMULINK, PID, Solver.

2.1 MATLAB, SIMULINK dan Solver

MATLAB atau yang biasa disebut dengan *Matrix Laboratory* adalah sebuah program yang digunakan untuk menganalisis dan memecahkan masalah matematis. MATLAB dikembangkan oleh *MathWorks Inc.* pada tahun 1984 [3]. MATLAB memiliki perangkat grafik yang cukup berguna dan dapat membuat gambar-gambar dalam 2D dan 3D, yaitu fungsi plot. Fungsi plot pada umumnya menggambarkan dalam bentuk *polyline* [4]. Dalam hal simulasi, MATLAB memiliki banyak fitur-fitur dan *plug-in* tambahan yang memudahkan untuk melakukan simulasi *modelling*. Salah satu *plug-in* tersebut adalah SIMULINK yang akan dibahas pada subbab berikut.

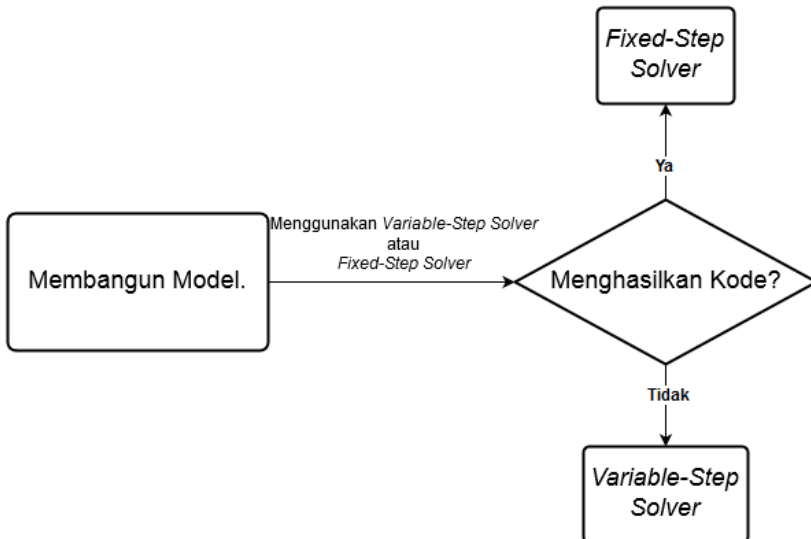
SIMULINK adalah *plug-in* tambahan dari MATLAB yang dikembangkan oleh *MathWorks Inc.* SIMULINK pada umumnya terdiri dari *block-block* diagram yang mewakili model dari proses suatu sistem [3]. SIMULINK merupakan program untuk menjalankan simulasi yang terintegrasi dengan MATLAB [3], sehingga melalui MATLAB script dapat dilakukan beberapa interaksi pada SIMULINK, seperti menambahkan *block*, mengatur parameter *block* tertentu, dll.

Dalam hal mensimulasikan model dari proses suatu sistem, terdapat perhitungan komputasi yang rumit. Oleh karena itu dibutuhkan penyelesaian untuk melakukan simulasi. Dalam hal ini SIMULINK dilengkapi dengan suatu *library* yang disebut *solver*.

Metode pada SIMULINK solver pada umumnya menggunakan ode (*Ordinary Differential Equation*) solver. ODE solver adalah penyelesaian terhadap persamaan *differential* dengan

menggunakan pendekatan numerik. Salah satu contohnya adalah ode45. Ode45 adalah ode *solver* dengan menggunakan metode Runge-Kutta Dormand-Prince Orde 4 dan 5, dimana orde kelima melakukan penghitungan estimasi *error* terhadap orde keempat [5].

Pada SIMULINK secara umum terbagi menjadi dua tipe *solver*, yaitu: *fixed-step solver*, dan *variable-step solver*. *Fixed-step solver* umumnya digunakan pada simulasi *modelling* yang mengandung banyak *switch*, sedangkan *variable-step solver* umumnya digunakan pada simulasi *modelling* yang murni mengandung model yang kontinu [5], dan juga salah satu perbedaan yang mendasar adalah apakah simulasi model yang dibangun perlu untuk dihasilkan menjadi *code* atau tidak, seperti pada Gambar 2.1 [5]. Jika perlu, maka *solver* yang cocok adalah *fixed-step solver*; jika tidak, maka *solver* yang cocok adalah *variable-step solver*.

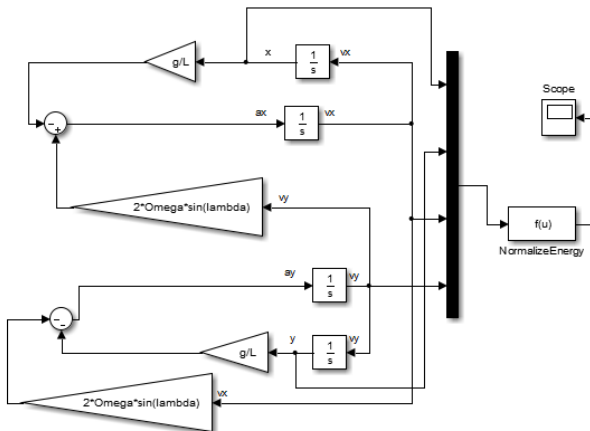


Gambar 2.1 Pemilihan Solver pada SIMULINK

Fixed-step solver adalah salah satu jenis solver yang ada pada SIMULINK yang dengan konstan *step size*. *Step size* adalah delta

waktu untuk menjalankan simulasi sampai akhir. Sedangkan *variable-step solver* adalah salah satu jenis *solver* pada SIMULINK yang dengan variatif menambahkan *step size* bergantung pada model dinamis dari prosesnya, tapi pada umumnya perubahan itu berdasarkan pada toleransi *error* yang telah di-*set* lebih dahulu.

Fixed-step solver terdiri dari 7 solver, yaitu: *fixed-step discrete solver*, ode1, ode2, ode3, ode4, ode5, ode8, ode14x [5]. *Fixed-step discrete solver* adalah *solver* untuk menyelesaikan pemodelan yang tidak ada unsur kontinu. Ode14x adalah *solver* untuk menyelesaikan permasalahan simulasi model yang *stiff*. Maksud dari model yang *stiff* adalah model yang memiliki dinamika yang terus menerus yang bervariasi perlahan dan cepat pada perubahan *step size*, contoh model *stiff* dapat dilihat pada Gambar 2.2. Sedangkan untuk ode1, ode2, ode3, ode4, ode5, dan ode8 adalah *solver* untuk model yang *non-stiff*. Untuk penggunaan ode1, ode2, ode3, ode4, ode5, dan ode8, umumnya digunakan berdasarkan kebutuhan sesuai orde derivasi model yang digunakan atau bisa juga dicoba satu-persatu dari ode1 sampai ode8 yang hasilnya paling mendekati dengan *variable-step solver* itu yang dipilih.



Gambar 2.2 Contoh Model Stiff : Foucault Pendulum Model [6]

Variable-step solver terdiri dari 8 solver, yaitu *variable-step discrete solver*, ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb [5]. *Variable-step discrete solver* adalah *solver* untuk menyelesaikan domain diskrit. Ode15s, ode23s, ode23t, dan ode23tb adalah *solver* untuk model yang *stiff*. Sedangkan ode45, ode23, dan ode113 adalah *solver* untuk model yang *non-stiff* atau penjelasan lebih tepatnya dapat dilihat pada Tabel 2.1 dan Tabel 2.2 [5].

Tabel 2.1 Karakteristik ode45, ode23, ode113 (Bagian Pertama)

No.	Solver	Tingkat Akurasi	Penjelasan
1.	Ode45	Sedang	Ode45 adalah ode solver dengan metode Runge-Kutta Dormand-Prince orde 4 dan 5. Secara umum, ode45 <i>solver</i> adalah yang terbaik untuk diterapkan sebagai percobaan pertama untuk sebagian besar model simulasi. Untuk alasan ini, ode45 adalah <i>solver default</i> untuk model dengan keadaan yang kontinu. Ode45 lebih akurat dan cepat dibandingkan dengan ode23. Jika komputasi ode45 berjalan pelan, maka model yang digunakan adalah model <i>stiff</i> dan perlu solver yang lebih cocok.
2.	Ode23	Rendah	Ode23 adalah ode <i>solver</i> dengan metode Runge-Kutta Bogacki-Shampine orde 2 dan 3. ode23 bisa lebih efisien daripada ode45 pada <i>crude error tolerances</i> dan pada adanya model <i>stiff</i> yang rendah.

Tabel 2.2 Karakteristik ode45, ode23, ode113 (Bagian Kedua)

3.	Ode113	Bervariasi. Dari rendah ke tinggi	Ode113 adalah ode <i>solver</i> dengan metode <i>multistep</i> Adams-Bashforth-Moulton PECE. Untuk masalah dengan <i>error</i> toleransi yang ketat atau untuk masalah komputasi intensif, <i>solver</i> Adams-Bashforth-Moulton Pece(ode113) bisa lebih efisien daripada ode45.
----	--------	---	--

2.2 PID CONTROLLER

PID Controller adalah salah satu controller yang paling sering digunakan pada dunia industri dan dioperasikan lebih dari 80% dari sistem kontrol yang ada di dunia. PID control sering digunakan pada sistem kontrol karena strukturnya yang sederhana dan performa yang dihasilkan bagus tanpa komputasi yang rumit [7].

Struktur PID Control dapat dijelaskan pada Gambar 2.3 [8]. PID *control* terdiri dari 3 *controller*, yaitu P *Controller*, I *Controller*, dan D *Controller*, yang dapat dilihat pada Persamaan 2.1 [7]:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (2.1)$$

Dimana:

K_p adalah proportional gain

K_i adalah integral gain

K_d adalah derivative gain

$u(t)$ adalah controller output

e adalah nilai error

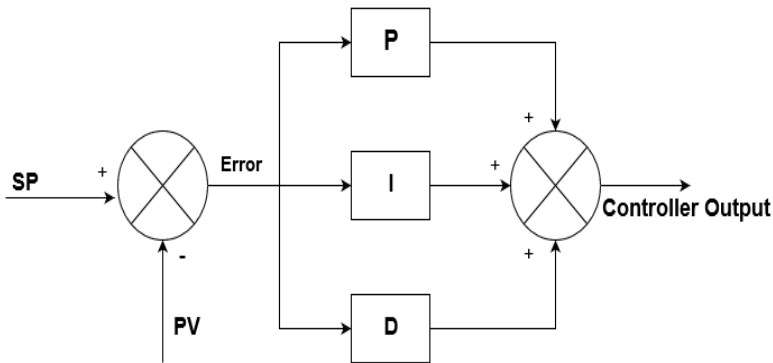
t adalah waktu atau waktu seketika

τ adalah variabel integrase; nilainya dari waktu 0 sampai waktu t .

Cara kerjanya cukup mudah, yaitu:

1. Nilai dari *Set Point* (SP) dikurangkan dengan nilai *Process Variable* (PV). *Set Point* ini adalah input untuk prosesnya. *Process Variable* adalah output dari tahap proses yang telah dijalankan. Output dari langkah ini adalah nilai error yang dihasilkan.
2. Nilai *error* yang dihasilkan masuk pada setiap *controller* dan diproses oleh setiap *controller*. Hasil dari tiap *controller* nantinya akan ditambahkan dan menghasilkan *controller output*. *Controller output* adalah nilai yang akan menjadi input untuk tahap proses.

Ketiga *controller* tersebut, *controller P*, *controller I*, *controller D*, masing-masing mempunyai karakteristik sendiri-sendiri. Hal ini nantinya diperlukan untuk *mensetting* parameter agar output sesuai dengan keadaan yang diinginkan atau bisa juga disebut *tuning*. Karakteristik tersebut dijelaskan dalam Tabel 2.3 [9].



Gambar 2.3 Struktur PID Controller

Tuning sering diartikan sebagai penyetelan parameter dari *controller* untuk mendapatkan respon yang terbaik, salah satunya adalah kestabilan sistem. *PID controller* juga perlu dilakukan *tuning* agar tanggapan dari *PID controller* berjalan dengan lebih baik.

Tabel 2.3 Karakteristik PID

Closed-loop Response	Rise-Time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_p	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

Setelah dipergunakannya PID *controller* pada dunia industri, banyak algoritma *tuning* yang berkembang. Yang paling terkenal diantaranya adalah metode ziegler-nichols. Berikut langkah-langkahnya [10]:

1. Matikan semua kontrol I, dan D dari PID.
2. Atur nilai parameter P menjadi 0. Lalu naikan secara perlahan parameter P sampai mencapai osilasi.
3. Nilai P yang sekarang, disebut P_{crit} , dengan waktu periode T_{crit} .
4. Ubah nilai parameter I dan D dengan nilai pada Tabel 2.4 [10].

Tabel 2.4 Tabel Tuning Ziegler-Nichols

<i>Controller</i>	K_p	K_I	K_D
P	$0.5 * P_{crit}$	-	-
PI	$0.45 * P_{crit}$	$0.85 * T_{crit}$	-
PID	$0.6 * P_{crit}$	$0.5 * T_{crit}$	$0.13 * T_{crit}$

Penerapan PID banyak dilakukan selain di bidang industri. Salah satunya pada penerapan *muscle model*. *Muscle model* adalah model matematika yang mewakili pergerakan dari otot tubuh manusia, contohnya adalah Hill's *muscle model* [11]. Untuk *muscle model*, yang akan dibahas kali ini, adalah *muscle model* untuk memodelkan pergerakan tangan manusia. Model pergerakan tangan manusia menghasilkan gaya terapan, yang mana mempunyai kemiripan dengan kinerja *PID Controller*, yaitu pada saat *PID Controller* mendorong *error* sehingga menghasilkan *error* yang dihasilkan relatif kecil. Gaya terapan inilah yang memberikan kesan seperti penekanan tangan, ketebalan pena, dan cara penggarisan. Kinerja *PID Controller* yang dimaksudkan adalah *P Controller*, yang menghitung *error*, *I Controller*, yang menghitung *error* sebelumnya sehingga *controller* melakukan aksi untuk mengurangi *error*, dan *D controller*, yang menghitung kemungkinan untuk *error* yang akan datang sehingga *controller* melakukan tindakan untuk mengurangi kemungkinan *error* tersebut. Penekanan *error* inilah yang menjadikan arsiran yang dihasilkan terkesan seperti adanya penekanan tangan, ketebalan pena, dan cara menggariskan. Sehingga model pergerakan tangan manusia dapat didekati dengan *PID Controller*.

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dijelaskan perancangan program yang dibuat. Perancangan akan dibagi menjadi empat proses utama, yaitu:

1. Membaca input 3D Referensi Trajectori dari *text file*.
2. Penggunaan PID *Controller* untuk menerapkan *muscle model*.
3. Pra-Simulasi untuk mengintegrasikan semua *block* agar simulasi dapat berjalan dan menampilkan hasil proses.
4. Simulasi untuk memilih *solver* dan menjalankan simulasi.

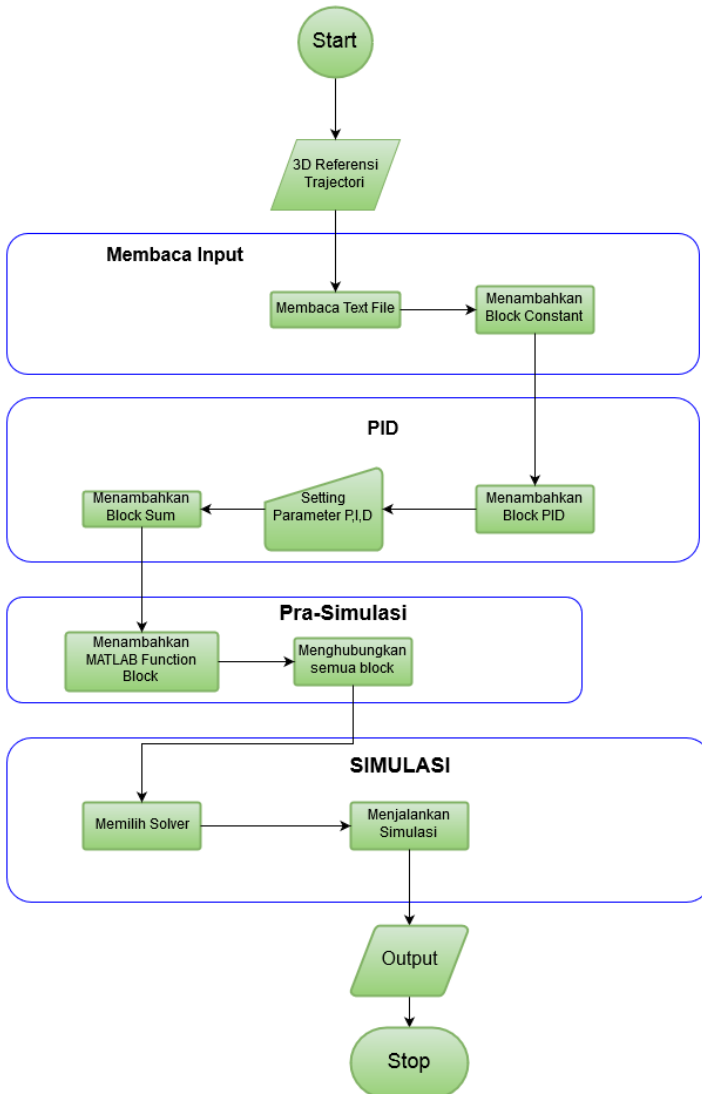
Pada bab ini akan dijelaskan gambaran umum setiap program utama dalam *flowchart* selanjutnya untuk penjelasan lebih detail akan disajikan dalam *pseudocode*.

3.1 Desain Metode Secara Umum

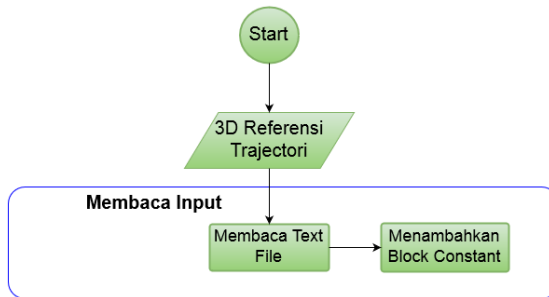
Metode ini terdiri dari tahap membaca input, penggunaan PID *Controller*, pra-simulasi, dan simulasi. Tahap membaca input dimulai dengan memilih *text file* untuk mendapatkan nilai referensi trajektori. Kemudian tahap selanjutnya adalah penggunaan PID *Controller* untuk menerapkan *muscle model*. Selanjutnya adalah pra-simulasi untuk mengintegrasikan semua *block* dan menampilkan hasil proses. Tahap terakhir adalah simulasi, untuk mensimulasikan pergerakan pena. Diagram alur dari proses aplikasi ditunjukkan pada Gambar 3.1.

3.2 Perancangan untuk Membaca Input

Dalam mengawali proses pengerjaan aplikasi, maka harus membaca input dahulu. Pada tugas akhir kali ini, tahap membaca input berupa 3D referensi trajektori dari *text file*. Gambaran umum dari proses membaca input digambarkan pada Gambar 3.2. Tabel 3.1 dan Tabel 3.2 berisi variabel-variabel yang akan digunakan pada proses ini. Tabel 3.3 berisi fungsi yang akan digunakan pada proses ini.



Gambar 3.1 Gambar Alur Proses Secara Umum



Gambar 3.2 Gambar Alur Proses Membaca Input

Tabel 3.1 Daftar Variabel yang Digunakan Pada *Pseudocode* Membaca Input (Bagian Pertama)

No.	Nama Variabel	Tipe	Penjelasan
1.	fileID	int	Penanda apakah file bisa dibuka atau tidak. Jika bisa, bernilai mulai dari 3. Jika tidak bisa, bernilai -1.
2.	rLine	string	Variabel yang menampung 1 baris hasil baca dari <i>file</i> yang sedang dibuka.
3.	BlockPath	string	Variabel yang berisi <i>path</i> dari <i>block</i> yang ingin ditambahkan.
4.	blockName	string	Variabel yang berisi nama <i>block</i> . Dengan ketentuan: 'NamaSistem/Var'. Var bisa diisi dengan nilai tertentu tapi harus unik.
5.	posl	int	Variabel yang berisi posisi atas dari <i>block</i> . Variabel ini berfungsi untuk menentukan posisi <i>block</i> pada lembar kerja SIMULINK.

Tabel 3.2 Daftar Variabel yang Digunakan Pada Pseudocode Membaca Input (Bagian Kedua)

6.	posJ	int	Variabel yang berisi posisi bawah dari <i>block</i> . Variabel ini berfungsi untuk menentukan posisi <i>block</i> pada lembar kerja SIMULINK.
7.	nInput	int	Jumlah variabel yang berisi nilai input.

Tabel 3.3 Daftar Fungsi yang Digunakan Pada Pseudocode Membaca Input

No.	Nama Fungsi	Penjelasan
1.	addBlock Constant	Fungsi untuk menambahkan <i>block constant</i> pada SIMULINK.

3.2.1 Membaca dari text file

Pada tugas akhir ini, input berupa *text file* dengan susunan tertentu, seperti pada Gambar 3.3. Di mana baris pertama berupa array letak posisi kamera untuk melihat objek. Baris kedua berupa limit dari x-axis. Baris ke-tiga berupa limit dari y-axis. Baris ke-empat berupa limit dari z-axis. Baris ke-lima berupa nilai untuk ketebalan pena. Baris ke-enam berupa array untuk nilai warna pena. Baris ke-tujuh dan seterusnya berupa array untuk referensi trajektori 3D. Untuk penjelasan lebih jelasnya dapat dilihat pada Gambar 3.4 dan Gambar 3.5 yang merupakan *pseudocode* dari proses membaca dari *text file*. Untuk isi *text file* dari objek yang telah dibuat, dapat dilihat pada Lampiran A.


```

[-60, 22] Camera View
[-2 2] Limit X-Axis
[-2 2] Limit Y-Axis
[-2 2] Limit Z-Axis

1.5| Ketebalan Pena

[0.375 0.375 0.375] Nilai Untuk Warna

[-1 -1 -1.25; -1 1 -1.25; -1 1 1.25;
[0.5 0.5 1.25; -0.5 0.5 1.25; -0.5 -0.5 1.25;
[0.5 -1 -0.5; -0.5 -1 -0.5; -0.5 -1 0.5;
[-1 0.5 -0.5; -1 0.5 0.75; -1 -0.5 0.75;
-0.5; -0.35 -0.13 -0.5; -0.375 0.5 -0.5;
-0.15 0.1 0.05; -0.25 -0.07 0.08; -0.25 -0.07 0.08;
[-0.5 -1.2 -1.25; -0.3 -1.075 -1.25;
[-0.35 -2.05 -1; -0.4 -2.5 -1.25; -0.4 -2.5 -1.25;
[-1.35 -1.95 -1; -1.625 -2.4 -1.25;

```

Referensi Trajektori

Gambar 3.3 Contoh Tampilan dari *Text File*

Masukan	Nama text file yang akan dibaca(variabel fileName)
Keluaran	Jumlah nilai Input(variabel arr_nInput)

Gambar 3.4 *Pseudocode* Membaca dari Text File (Bagian Pertama)

```

1.  fileID ← File_open(fileName)
2.  nInput ← 0
3.  posI ← 30
4.  posJ ← 60
5.  While(not EndOfFile(fileID))
6.      rLine ← fileReadLine(fileID)
7.      if length(rLine) > 0
8.          addBlockConstant(rLine,blockName,
                             posI,posJ)
9.      end
10. end while

```

Gambar 3.5 Pseudocode Membaca dari Text File (Bagian Kedua)

3.2.2 Menambahkan Block Constant

Program menambahkan *block constant* merupakan program yang melakukan penambahan *block* yang bertipe matrix pada SIMULINK. Penambahan *block* ini bertujuan agar dapat diproses di dalam SIMULINK. *Pseudocode* proses penambahan *block* ditunjukkan pada Gambar 3.6.

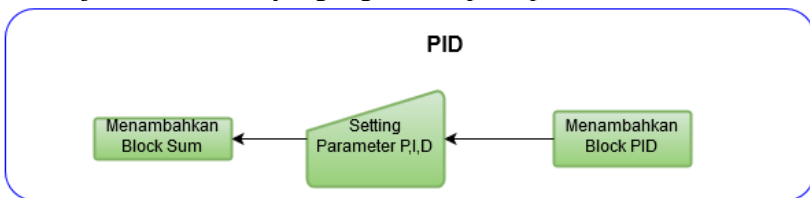
Masukan	Array hasil dari membaca text file(variabel arr_in), Variabel yang berisi posisi kiri dari block(variabel posI), variabel yang berisi posisi kanan dari block (variabel posJ) dan Variabel yang berisi nama block (variabel blockName)
Keluaran	Block Constant pada SIMULINK
<pre> 1. BlockPath ← 'simulink/Sources/Constant' 2. add_block(BlockPath,blockName) 3. set_param(blockName,'Value',arr_in) 4. set_param(blockName,'Position',[130 posI 160 posJ]) </pre>	

Gambar 3.6 Pseudocode Penambahan Block Constant

3.3 Perancangan PID

Salah satu proses utama yang dilakukan pada tugas akhir ini adalah perancangan PID. Trajektori referensi yang telah dimasukkan pada *block constant* nantinya akan diproses oleh *PID Controller* untuk menghasilkan *muscle model* agar dapat dilalui dari pena.

Penjelasan proses perancangan PID akan dijelaskan lebih detail dalam bentuk *pseudocode*. Adapun gambaran langkah secara umum seperti yang ditunjukkan pada Gambar 3.7. Tabel 3.4 menunjukkan variable yang digunakan pada proses PID.



Gambar 3.7 Gambar Alur Proses PID

Tabel 3.4 Daftar Variabel yang Digunakan Pada *Pseudocode* PID

No.	Nama Variabel	Tipe	Penjelasan
1.	BlockPath	string	Variabel yang berisi <i>path</i> dari <i>block</i> yang ingin ditambahkan.
2.	blockName	string	Variabel yang berisi nama block. Dengan ketentuan: 'NamaSistem/Var'. Var bisa diisi dengan nilai tertentu tapi harus unik.
3.	posI	int	Variabel yang berisi posisi atas dari <i>block</i> .
4.	posJ	int	Variabel yang berisi posisi bawah dari <i>block</i> .

3.3.1 Menambahkan Block PID

Program menambahkan *block* PID merupakan program yang melakukan penambahan *block* PID pada SIMULINK. Penambahan *block* ini bertujuan untuk membuat *block* PID sebanyak yang diperlukan. Tentunya *block* PID yang digunakan mempunyai struktur seperti yang telah dijelaskan pada subab 2.2. *Pseudocode* proses penambahan *block* ditunjukkan pada Gambar 3.8.

Masukan	Variabel yang berisi posisi atas dari block (variabel posI), variabel yang berisi posisi bawah dari block (variabel posJ) dan Variabel yang berisi nama block (variabel blockName)
Keluaran	Block PID pada SIMULINK
<pre>1. BlockPath ← 'simulink/Continuous/PID Controller' 2. add_block(BlockPath,blockName) 3. set_param(blockName,'Position',[350 posI 390 posJ])</pre>	

Gambar 3.8 *Pseudocode* Penambahan Block PID

3.3.2 Setting Parameter PID

Program setting parameter PID merupakan program yang men-*setting* parameter *control* PID pada *block PID* SIMULINK. Proses *setting* parameter ini bertujuan agar proses kontrol PID berjalan stabil pada SIMULINK seperti yang telah dijelaskan pada subbab 2.2, yaitu dengan melakukan *tuning* pada parameter PID. *Pseudocode* proses *setting* parameter ditunjukkan pada Gambar 3.9.

Masukan	Nilai parameter 'P' (variabel Kp), Nilai parameter 'I' (variabel Ki), Nilai parameter 'D' (variabel Kd) dan Variabel yang berisi nama block (variabel blockName)
Keluaran	Perubahan Parameter Block PID pada SIMULINK
<pre> 1. set_param(blockName,'P',Kp); 2. set_param(blockName,'I',Ki); 3. set_param(blockName,'D',Kd); </pre>	

Gambar 3.9 Pseudocode Setting Parameter PID

3.3.3 Penambahan *Block Sum*

Program menambahkan *block sum* merupakan program yang melakukan penambahan *block sum* pada SIMULINK. Penambahan *block* ini bertujuan agar stuktur PID mempunyai struktur PID pada umumnya. *Pseudocode* proses penambahan *block* ditunjukkan pada Gambar 3.10.

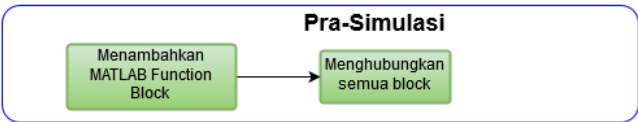
Masukan	Variabel yang berisi nama block (variabel blockName), Variabel yang berisi posisi atas dari block (variabel posI), variabel yang berisi posisi bawah dari block (variabel posJ)
Keluaran	<i>Block Sum</i> pada SIMULINK
<pre> 1. BlockPath←'simulink/Math Operations/Sum' 2. add_block(BlockPath,blockName) 3. set_param(blockName,'Inputs',' +-') 4. set_param(blockName,'Position',[235 posI 255 posJ]) </pre>	

Gambar 3.10 Pseudocode Penambahan *Block Sum*

3.4 Perancangan Pra-Simulasi

Pada proses kali ini, yakni Pra-Simulasi, bertujuan sebagai proses bantuan agar proses simulasi dapat dijalankan dengan baik. Proses ini dibutuhkan untuk dapat melihat hasil visualisasi dari proses sebelumnya pada saat simulasi dan untuk menghubungkan semua *block* pada lembar kerja SIMULINK agar simulasi dapat berjalan. Setelah semua penambahan *block-block* pada proses sebelumnya, *block-block* tersebut butuh dihubungkan dan untuk dapat melihat hasilnya, dibutuhkan *block* tambahan yaitu MATLAB *Function Block*.

Penjelasan proses Pra-Simulasi akan dijelaskan lebih detail dalam bentuk *pseudocode*. Adapun gambaran langkah secara umum seperti yang ditunjukkan pada Gambar 3.11. Tabel 3.5 dan Tabel 3.6 menunjukkan variabel yang digunakan pada proses Pra-Simulasi.



Gambar 3.11 Gambar Alur Proses Pra-Simulasi

Tabel 3.5 Daftar Variabel yang Digunakan Pada *Pseudocode* Pra-Simulasi (Bagian Pertama)

No.	Nama Variabel	Tipe	Penjelasan
1.	SName	string	Variabel yang berisi nama <i>system/file</i> SIMULINK.
2.	CName	string	Variabel yang berisi nama <i>block constant</i> yang dikontrol.
3.	CNName	string	Variabel yang berisi nama <i>block constant</i> yang tidak dikontrol.
4.	PIDN	string	Variabel yang berisi nama <i>block</i> PID.

Tabel 3.6 Daftar Variabel yang Digunakan Pada *Pseudocode* Pra-Simulasi (Bagian Kedua)

5.	SumName	string	Variabel yang berisi nama block <i>Sum</i> .
6.	oPort	string	Variabel yang berisi string: “/jumlahPort”. Untuk <i>block</i> PID, jumlah Output Portnya adalah 1; untuk <i>block constant</i> , jumlah Output Portnya adalah 1; untuk <i>block sum</i> , jumlah Output Portnya adalah 1;
7.	iPort	string	Variabel yang berisi string: “/jumlahPort”. Untuk <i>block</i> PID, jumlah Portnya adalah 1; untuk <i>block constant</i> , jumlah Portnya adalah 1; untuk <i>block sum</i> , jumlahPortnya adalah 2; untuk <i>MATLAB Function</i> , jumlahPortnya adalah sesuai nilai dari variabel nCount.
8.	MFunc	string	Variabel yang berisi nama <i>block MATLAB Function</i> .
9.	nInput	int	Jumlah variabel yang berisi nilai input.

3.4.1 Menambahkan MATLAB Function Block

Proses menambahkan *MATLAB Function Block* merupakan proses menambahkan *block MATLAB Function* pada SIMULINK dengan *drag and drop* dari *library* SIMULINK. Penambahan *block* ini bertujuan untuk proses yang selama ini terjadi dapat terlihat. Proses penambahan *block MATLAB Function* dapat dilakukan seperti berikut:

- 1. Buka SIMULINK
- 2. Pilih View → Library Browser
- 3. Pilih User's Define Function → MATLAB Function
- 4. Drag and Drop pada halaman SIMULINK
- 5. Ganti isi fungsi dengan banyak input dan tambahkan fungsi plot3

3.4.2 Menghubungkan Block

Program menghubungkan *block* merupakan program yang melakukan penambahan garis penghubung pada setiap *block* pada SIMULINK. Penambahan block ini bertujuan agar setiap *block* pada SIMULINK dapat terintegrasi. *Pseudocode* proses penambahan *block* ditunjukkan pada Gambar 3.12.

Masukan	Nama System(variabel SystemName), Nama Block Constant (variabel ConstantName), Nama Block PID(variabel PIDName), Nama Block Sum(variabel SumName), Jumlah variabel yang input(variabel nInput)
Keluaran	Block pada SIMULINK terintegrasi
1. add_line(SName,CNName+oPort,MFunc+iPort) 2. add_line(SName,PIDN+oPort,SumName+iPort) 3. add_line(SName,CName+oPort,SumName+iPort) 4. add_line(SName,SumName+oPort,PIDN+iPort) 5. add_line(SName,PIDN+oPort,MFunc+iPort)	

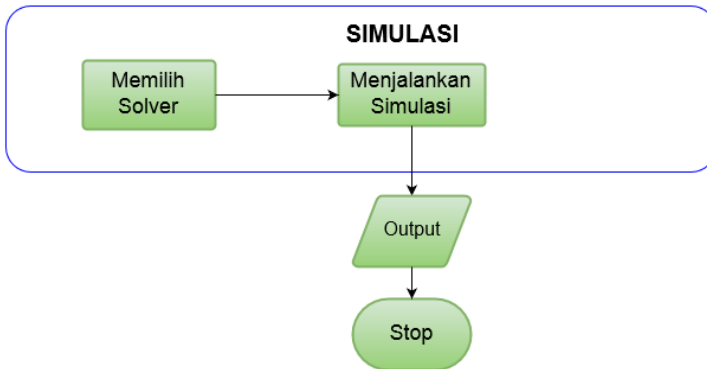
Gambar 3.12 Pseudocode Penambahan Block Sum

3.5 Perancangan Simulasi

Salah satu proses utama yang dilakukan pada tugas akhir ini adalah perancangan Simulasi. Setelah semua *block* ditambahkan dan dihubungkan, sentuhan terakhir adalah menjalankan simulasi yang sesuai. Pada tahapan ini, dilakukan dua tahapan proses yang

terhimpun ke dalam subproses-subproses, yaitu memilih *solver* yang sesuai, menjalankan simulasi.

Penjelasan proses Simulasi akan dijelaskan lebih detail dalam bentuk *step-by-step*. Adapun gambaran langkah secara umum seperti yang ditunjukkan pada Gambar 3.13.



Gambar 3.13 Gambar Alur Proses Simulasi

3.5.1 Memilih *Solver*

Proses memilih *solver* merupakan proses yang menentukan bagaimana setiap persamaan yang ada dalam SIMULINK dapat diselesaikan dengan baik. Proses pemilihan *solver* dapat dilakukan sebagai berikut:

1. Buka sistem atau *file* SIMULINK
2. Pilih Model Configuration Parameter
3. Pilih Solver yang diinginkan

3.5.2 Menjalankan Simulasi

Proses ini adalah proses terakhir, yaitu menjalankan simulasi. Proses menjalankan simulasi dapat dilakukan sebagai berikut:

1. Buka sistem atau *file* SIMULINK
2. Pilih Run

(halaman ini sengaja dikosongkan)

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Di dalamnya mencakup proses penerapan dan pengimplementasian algoritma, dan antar muka yang mengacu pada rancangan yang telah dibahas sebelumnya.

4.1 Lingkungan Implementasi

Lingkungan implementasi dari tugas akhir dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

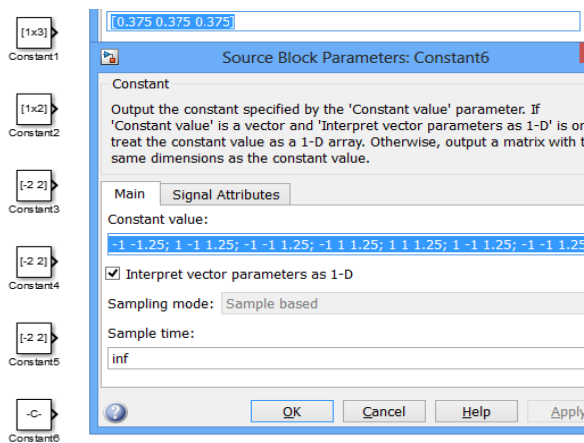
Perangkat Keras	Prosesor: Intel(R) Core(TM) i3-3240 CPU @ 3.40GHz(4 CPUs), ~ 3.4GHz Memori: 2 GB
Perangkat Lunak	Sistem Operasi: Microsoft Windows 8.0 Pro 64-bit Perangkat Pengembang: MATLAB R2013a Simulink Version : Simulink 8.1

4.2 Implementasi

Pada bagian ini akan dijelaskan implementasi setiap subbab yang terdapat pada bab sebelumnya yaitu bab analisis dan perancangan. Bagian implementasi ini juga akan dibagi menjadi 4 bagian, yaitu bagian membaca input, bagian PID, bagian pra-simulasi dan bagian simulasi.

4.2.1 Implementasi Membaca Input

Pada bagian ini akan dijelaskan implementasi dari membaca input. Implementasi akan dijelaskan mulai dari membaca *text file*. Setelah itu dilanjutkan dengan menambahkan block constant pada SIMULINK. Adapun hasil dari proses ini dapat ditunjukkan pada Gambar 4.1.



Gambar 4.1 Hasil dari Proses Membaca Input

4.2.1.1 Membaca Text File

Masukan dari program membaca *text file* berupa *text file* yang berisi trajektori referensi 3D dengan ketentuan tertentu yang sudah dijelaskan pada subbab 3.2.1. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.1. Implementasi dalam matlab ditunjukkan pada Kode Sumber 4.1 dan Kode Sumber 4.2.

1	<code>function nInput = ReadFile(fileName)</code>
2	<code>fileID = fopen(fileName);</code>
3	<code>nInput=0;</code>
4	<code>posI=30;</code>
5	<code>posJ=60;</code>
6	<code>while ~feof(fileID)</code>
7	<code> rLine = fgetl(fileID);</code>
8	<code> if (size(rLine)>0)</code>
9	<code> nCount=nCount+1;</code>

Kode Sumber 4.1 Implementasi Proses Membaca *Text File*
(Bagian Pertama)

10	<code>blockName=sprintf('Coba/ Constant%d',nCount);</code>
11	<code>addBlockConstant(rLine, blockName,posI,posJ);</code>
12	<code>posI=posI+75;</code>
13	<code>posJ=posJ+75;</code>
14	<code>end;</code>
15	<code>end;</code>
16	<code>fclose(fileID);</code>
17	<code>end</code>

Kode Sumber 4.2 Implementasi Proses Membaca *Text File* (Bagian Kedua)

4.2.1.2 Menambahkan *Block Constant*

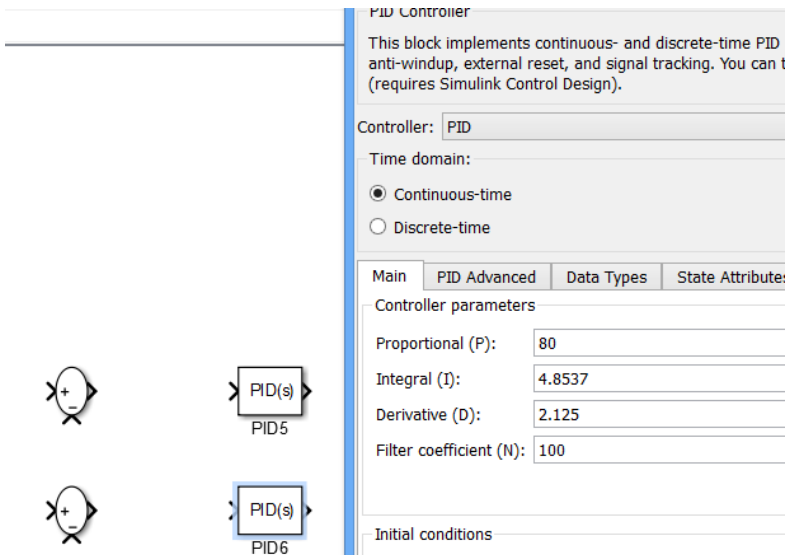
Masukan dari program menambahkan *block constant* berupa nilai hasil baca dari *text file* dan beberapa tambahan lain, seperti posisi atas *block*, posisi bawah *block*, nama *block* yang akan ditambahkan ke SIMULINK. Nilai hasil baca dari *text file* tersebut akan menjadi nilai dari *block constant*, posisi atas dan bawah *block* nantinya akan menentukan posisi dari *block* tersebut sedangkan posisi kiri dan kanan sudah ditentukan dari awal yaitu 130 dan 160. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.2. Implementasi dalam matlab ditunjukkan pada Kode Sumber 4.3.

1	<code>function addBlockConstant(arr_in,blockName, posI, posJ)</code>
2	<code>BlockPath='simulink/Sources/Constant';</code>
3	<code>add_block(BlockPath,blockName);</code>
4	<code>set_param(blockName,'Value',arr_in);</code>
5	<code>set_param(blockName,'Position',[130 posI 160 posJ]);</code>
6	<code>end</code>

Kode Sumber 4.3 Implementasi Proses Membaca *Text File*

4.2.2 Implementasi PID

Pada bagian ini akan dijelaskan implementasi dari PID. Implementasi akan dijelaskan mulai dari menambahkan *block* PID. Setelah itu dilanjutkan dengan setting parameter *block* PID. Selanjutnya dilanjutkan dengan menambahkan *block sum*. Adapun hasil dari proses ini dapat ditunjukkan pada Gambar 4.2.



Gambar 4.2 Hasil dari Proses PID

4.2.2.1 Menambahkan *Block* PID

Masukan dari program menambahkan *block* PID berupa posisi atas *block*, posisi bawah *block*, nama *block* yang akan ditambahkan ke SIMULINK. Posisi atas dan bawah *block* nantinya akan menentukan posisi dari *block* tersebut sedangkan posisi kiri dan kanan sudah ditentukan dari awal yaitu 350 dan 390. Implementasi ini merupakan implementasi *pseudocode* pada

subbab 3.3.1. Implementasi dalam matlab ditunjukkan pada Kode Sumber 4.4.

1	<code>function addBlockPID(blockName, posI, posJ)</code>
2	<code>BlockPath = 'simulink/Continuous/PID</code> <code>Controller';</code>
3	<code>add_block(BlockPath,blockName);</code>
4	<code>set_param(blockName,'Position',[350 posI 390</code> <code>posJ]);</code>
5	<code>End</code>

Kode Sumber 4.4 Implementasi Proses Menambahkan *Block* PID

4.2.2.2 *Setting* Parameter PID

Masukan dari program *setting* parameter PID berupa nama block PID yang akan di-*setting*, nilai parameter untuk P, nilai parameter untuk I, nilai parameter untuk D. Implementasi ini merupakan implementasi pseudocode pada subbab 3.3.2. Setelah melalui *tuning*, seperti yang sudah dijelaskan pada subbab 2.2, nilai parameter yang dipakai adalah P=80, I=4.8537, dan D=2.125. Implementasi dalam matlab ditunjukkan pada Kode Sumber 4.5.

1	<code>function SettingParamPID(blockName, Kp, Ki,</code> <code>Kd)</code>
2	<code>set_param(blockName,'P',Kp);</code>
3	<code>set_param(blockName,'I',Ki);</code>
4	<code>set_param(blockName,'D',Kd);</code>
5	<code>End</code>

Kode Sumber 4.5 Implementasi Proses *Setting* Parameter PID

4.2.2.3 Menambahkan *Block Sum*

Masukan dari program menambahkan *block sum* berupa posisi atas *block*, posisi bawah *block*, nama *block* yang akan ditambahkan ke SIMULINK. Posisi atas dan bawah *block* nantinya akan menentukan posisi dari *block* tersebut sedangkan posisi kiri dan kanan sudah ditentukan dari awal yaitu 235 dan 255.

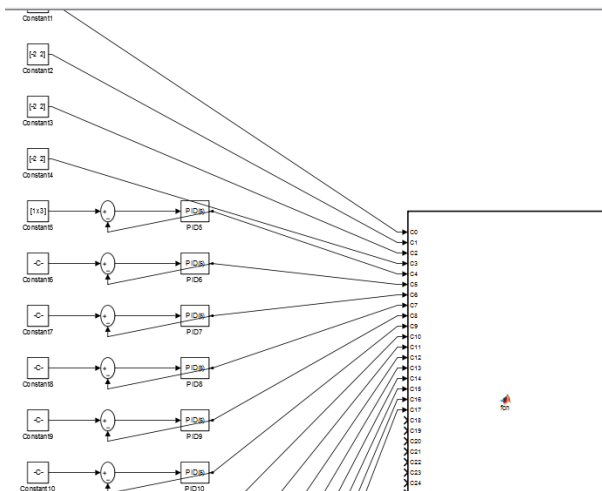
Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.3.3. Implementasi dalam matlab ditunjukkan pada Kode Sumber 4.6.

1	<code>function addSumBlocks(blockName,posI,posJ)</code>
2	<code>BlockPath='simulink/Math Operations/Sum';</code>
3	<code>add_block(BlockPath,blockName);</code>
4	<code>set_param(blockName,'Inputs',' +-');</code>
5	<code>set_param(blockName,'Position',[235 posI 255 posJ]);</code>
6	<code>End</code>

Kode Sumber 4.6 Implementasi Proses Menambahkan *Block Sum*

4.2.3 Implementasi Pra-Simulasi

Pada bagian ini akan dijelaskan implementasi dari pra-simulasi. Implementasi akan dijelaskan mulai dari menambahkan *block* MATLAB *Function*. Setelah itu dilanjutkan dengan menghubungkan semua *block*. Adapun hasil dari proses ini dapat ditunjukkan pada Gambar 4.3.

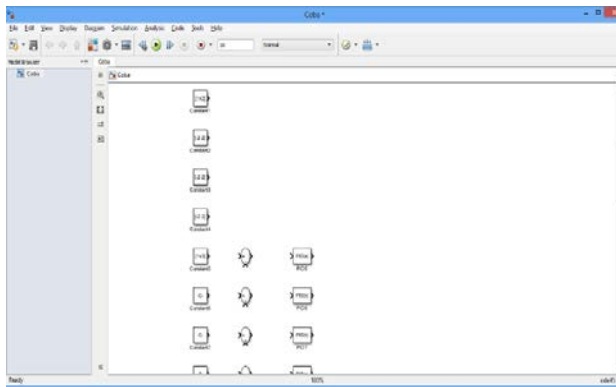


Gambar 4.3 Hasil dari Proses Pra-Simulasi

4.2.3.1 Menambahkan *MATLAB Function*

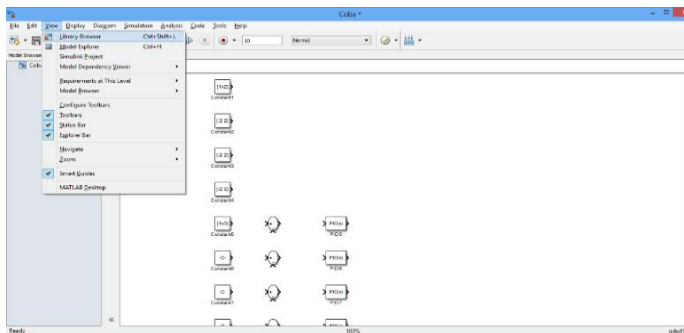
Pada proses ini dilakukan *step-by-step* di dalam SIMULINK. Karena sejauh ini tidak bisa merubah isi dalam dari *MATLAB functions* melalui *MATLAB Scripts*. Implementasi ini merupakan implementasi penjelasan langkah pada subbab 3.4.1. Implementasi ini akan ditunjukkan seperti berikut:

1. Membuka SIMULINK.



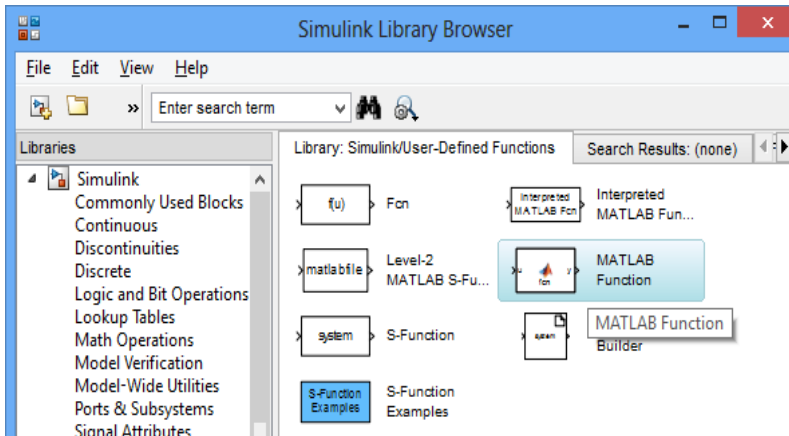
Gambar 4.4 Membuka SIMULINK

2. Membuka *Library Browser* yang terletak pada *View* di bar navigasi.



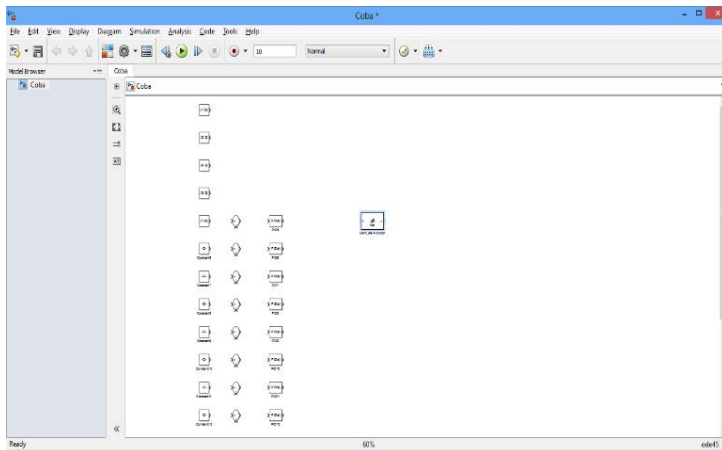
Gambar 4.5 Membuka Library Browser

3. Memilih *MATLAB Function* yang terletak pada tab Simulink → *User-Defined Function*.



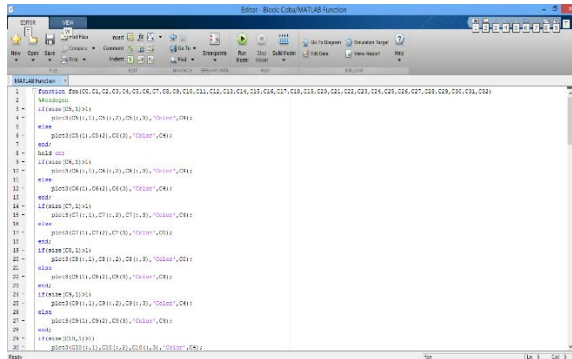
Gambar 4.6 Memilih MATLAB Function

4. Men-drag *MATLAB Function* pada lembar kerja SIMULINK.



Gambar 4.7 Men-drag MATLAB Function pada lembar kerja SIMULINK

5. Mengubah isi *MATLAB Function* seperti pada Gambar 4.8. Untuk lebih lengkapnya, dapat dilihat pada Lampiran B.



Gambar 4.8 Mengubah isi MATLAB Function

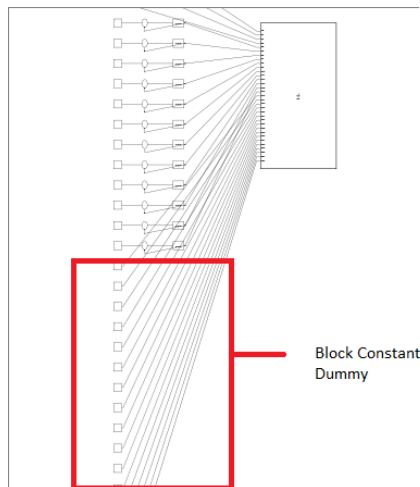
4.2.3.2 Menghubungkan Semua *Block*

Masukan dari program menghubungkan semua *block* berupa nama sistem yang digunakan, nama *block constant* yang dikontrol, nama *block constant* yang tidak dikontrol, nama *block PID*, nama *block sum*, jumlah *port* baik untuk input maupun untuk output dari setiap *block*. Nama sistem adalah lembar kerja SIMULINK yang dipakai. Nama *block constant* yang dikontrol adalah *block constant* yang nantinya akan diproses oleh PID. Nama *block constant* yang tidak dikontrol adalah *block constant* yang tidak akan diproses oleh PID. Nama *block PID* adalah nama yang diberikan pada *block PID* pada lembar kerja SIMULINK. Begitu pula dengan nama *block sum* adalah nama yang diberikan pada *block sum* pada lembar kerja SIMULINK. Untuk jumlah *port* adalah nilai seperti yang dijelaskan pada table 3.4. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.4.2. Implementasi dalam matlab ditunjukkan pada Kode Sumber 4.7.

Kode Sumber 4.8 Implementasi Penambahan Block Constant Dummy (Bagian Pertama)

6	<code>set_param(blockName, 'VectorParams1D','off');</code>
7	<code>MFuncPort=sprintf('MATLAB Function/%d',x);</code>
8	<code>CNName=sprintf('Constant%d/1',x);</code>
9	<code>add_line(SName,CNName,MFuncPort);</code>
10	<code>posI=posI+75;</code>
11	<code>posJ=posJ+75;</code>
12	<code>end;</code>
13	<code>end;</code>

Kode Sumber 4.9 Implementasi Penambahan Block Constant Dummy (Bagian Kedua)

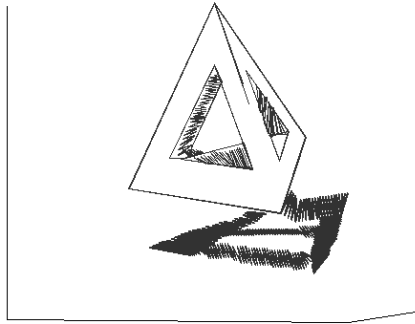


Gambar 4.9 Menambah block constant dummy

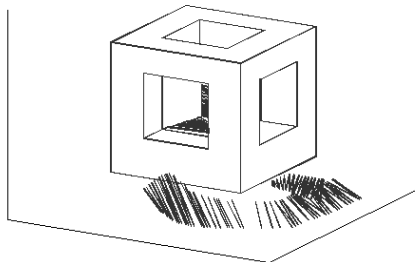
4.2.4 Implementasi Simulasi

Pada bagian ini akan dijelaskan implementasi dari simulasi. Implementasi akan dijelaskan mulai dari memilih solver. Setelah itu dilanjutkan menjalankan simulasi. Adapun hasil dari

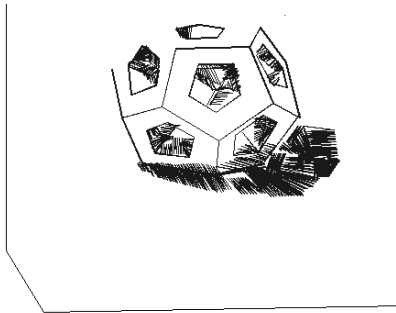
proses ini untuk input *Segitiga.txt* dapat ditunjukkan pada Gambar 4.10, untuk input *Cube.txt* dapat ditunjukkan pada Gambar 4.11, dan untuk input *Segilima.txt* dapat ditunjukkan pada Gambar 4.12. Untuk implementasi sistem dari objek yang telah dibuat, dapat dilihat pada Lampiran C.



**Gambar 4.10 Hasil Dari Proses Simulasi Dengan Input
*Segitiga.txt***



**Gambar 4.11 Hasil Dari Proses Simulasi Dengan Input
*Cube.txt***

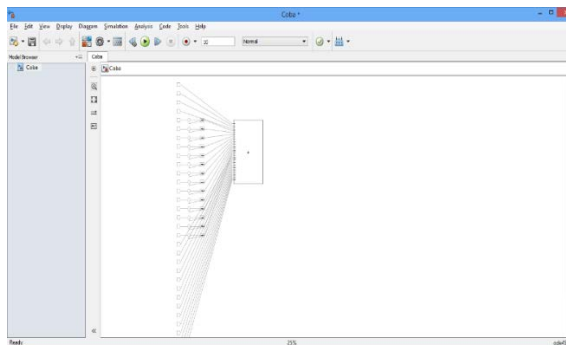


Gambar 4.12 Hasil Dari Proses Simulasi Dengan Input Segilima.txt

4.2.4.1 Memilih Solver

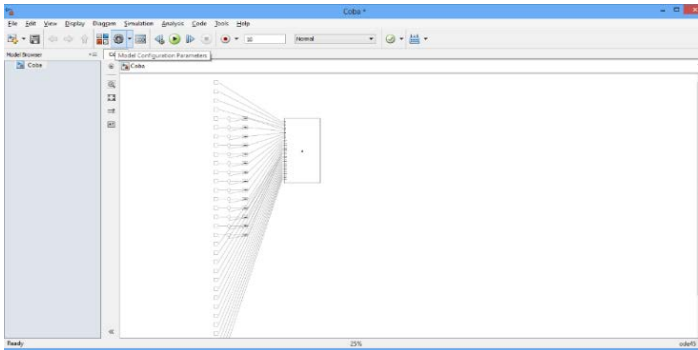
Pada proses ini, memilih dilakukan *step-by-step* di dalam SIMULINK. Implementasi ini merupakan implementasi penjelasan langkah pada subbab 3.5.1. Implementasi ini akan ditunjukkan seperti berikut:

1. Membuka SIMULINK tampilan akan seperti Gambar 4.13.



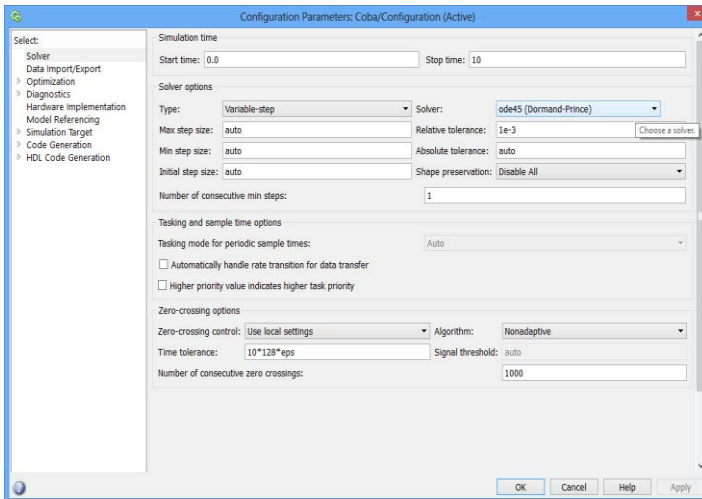
Gambar 4.13 Membuka SIMULINK

- Memilih Model Configuration Parameter seperti pada Gambar 4.14.



Gambar 4.14 Memilih Model Configuration Parameter

- Memilih solver seperti pada Gambar 4.15.



Gambar 4.15 Memilih solver

4.2.4.2 Menjalankan Simulasi

Pada proses ini, memilih dilakukan *step-by-step* di dalam SIMULINK. Implementasi ini merupakan implementasi pada subbab 3.5.2, yaitu membuka file SIMULINK, lalu men-*click run*.

(halaman ini sengaja dikosongkan)

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba serta analisis setiap pengujian.

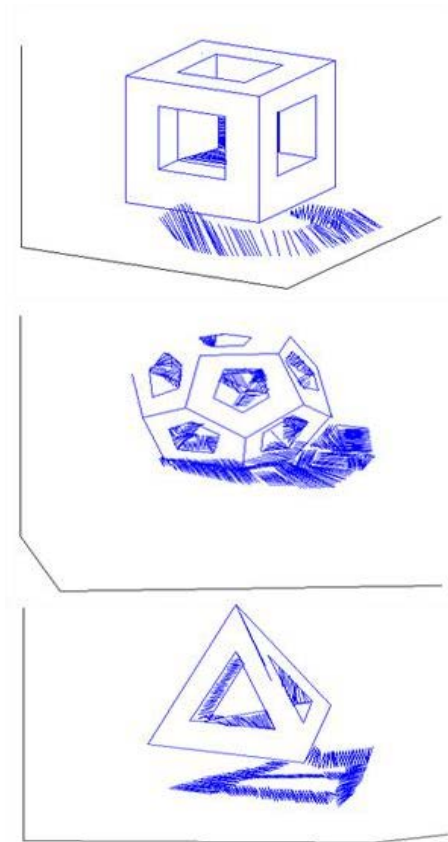
5.1 Lingkungan Uji Coba

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi parameter PID dan waktu proses simulasi pada tugas akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak ini dicantumkan seperti berikut:

1. Perangkat keras
 - a. Prosesor : Intel® Core™ i3-3240 CPU (3.40 GHz)
 - b. Memory (RAM) : 2,00 GB
 - c. Graphic Card : Intel® HD Graphics
 - d. Tipe Sistem : Sistem operasi 64-bit
 - e. Jenis Device : PC
2. Perangkat lunak
 - a. Sistem operasi : Windows 8.0 Professional
 - b. IDE : MATLAB 8.1.0 (R2013a), Simulink 8.1.

5.2 Data Uji Coba

Data yang digunakan pada uji coba ini adalah tiga referensi trajektori 3D. Tiga referensi trajektori 3D tersebut ditulis dalam *text file*, dan nama *file*-nya adalah *Cube.txt*, *Segilima.txt*, *Segitiga.txt*. Untuk hasil gambar dari trajektori referensi dapat dilihat pada Gambar 5.1.



Gambar 5.1 Visualisasi 3D Trajektori Referensi Untuk Diuji

5.3 Uji Coba Parameter PID

Pada uji coba kali ini akan dilakukan pengujian terhadap parameter PID. Uji coba parameter PID ini dilakukan dengan tujuan untuk mengetahui kestabilan sistem pada parameter PID yang berbeda.

5.3.1 Skenario Uji Coba Parameter PID

Uji coba ini dilakukan dengan mengubah parameter dari P , I , dan D dengan waktu simulasi 10 *seconds* dan solver ode113. Skenario untuk tiap parameter PID adalah sebagai berikut:

1. Setiap objek diuji dengan parameter $P=1$, $I=1$, dan $D=1$. Nilai ini adalah nilai dasar atau nilai awal untuk penggunaan PID.
2. Setiap objek diuji dengan parameter $P=80$, $I=4.8537$, dan $D=2.125$. Nilai ini adalah nilai yang didapatkan setelah proses *tuning*, seperti yang sudah dijelaskan pada subbab 4.2.2.2.

Hasil yang diharapkan dari pengujian ini adalah menunjukkan adanya sistem yang stabil dengan parameter tertentu.

5.3.2 Hasil Pengujian dan Evaluasi Parameter PID Skenario I

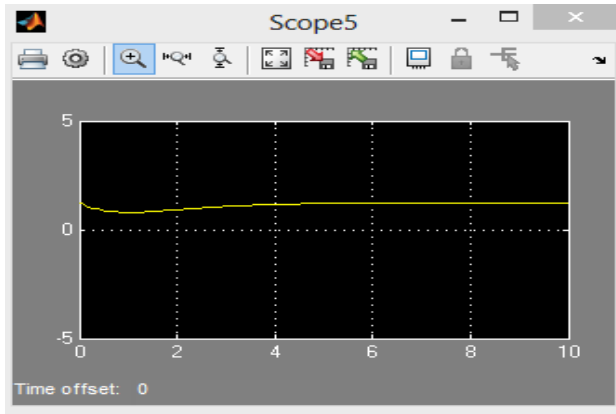
Pada skenario ini, setiap objek dijalankan dengan ditambahkan *block scope* dari simulink untuk melihat kestabilan sistem. Parameter yang diujikan adalah parameter P, I, D dari PID *Control*. Hasil dapat dilihat pada Gambar 5.2, Gambar 5.3, dan Gambar 5.4. Hasil yang didapat, baik objek dengan input *Cube.txt*, objek dengan input *Segilima.txt*, dan objek dengan input *Segitiga.txt*, menunjukkan sistem baru dapat berjalan stabil pada waktu 5 *seconds*.

Hasil tersebut menunjukkan bahwa selama proses simulasi berlangsung terdapat perubahan yang relatif signifikan setelah proses PID. Jadi untuk skenario ini hasilnya kurang bagus.

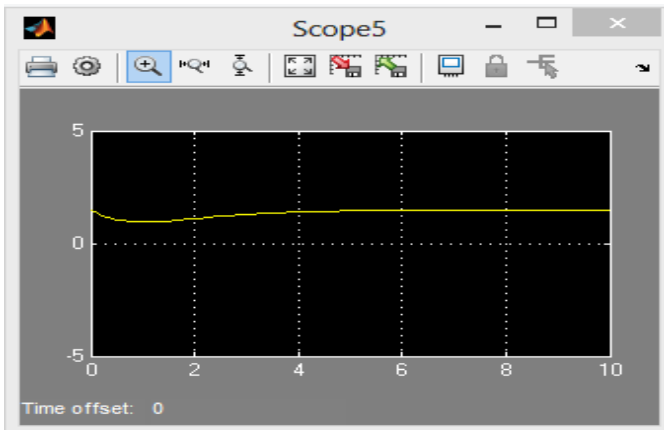
5.3.3 Hasil Pengujian dan Evaluasi Parameter PID Skenario II

Pada skenario ini, setiap objek dijalankan dengan ditambahkan *block scope* dari simulink untuk melihat kestabilan sistem. Parameter yang diujikan adalah parameter P, I, D dari PID *Control*. Hasil dapat dilihat pada Gambar 5.5, Gambar 5.6, dan Gambar 5.7. Hasil yang didapat, baik objek dengan input *Cube.txt*,

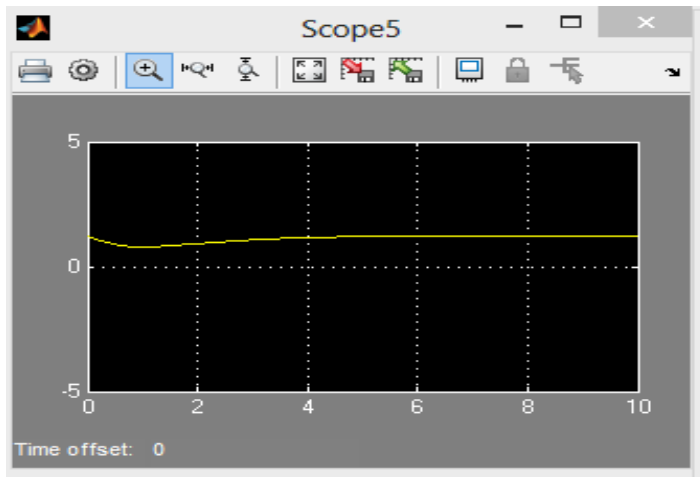
objek dengan input *Segilima.txt*, dan objek dengan input *Segitiga.txt*, menunjukkan sistem baru dapat berjalan stabil pada waktu 0 seconds.



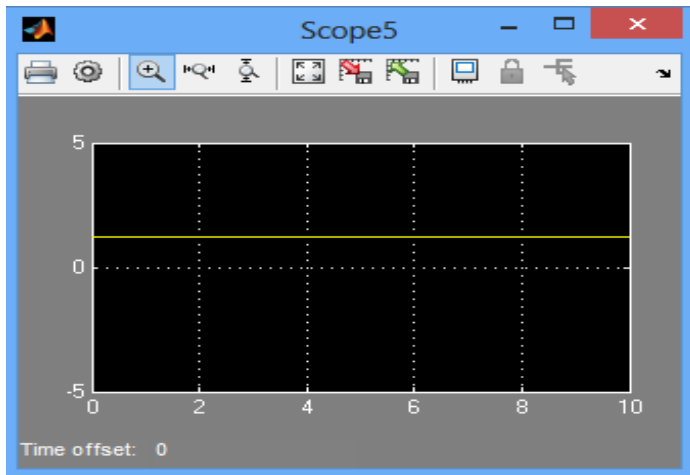
Gambar 5.2 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Kubus



Gambar 5.3 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Segilima

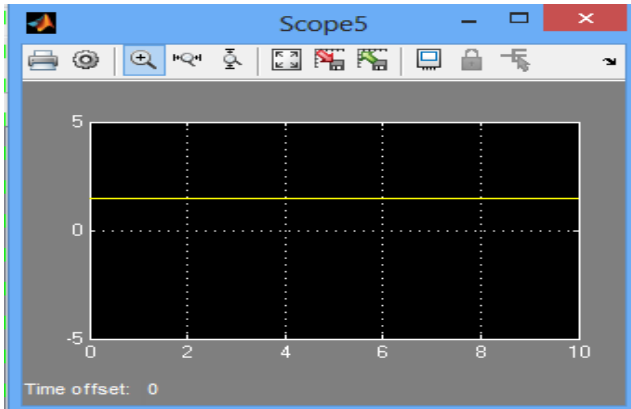


Gambar 5.4 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Segitiga

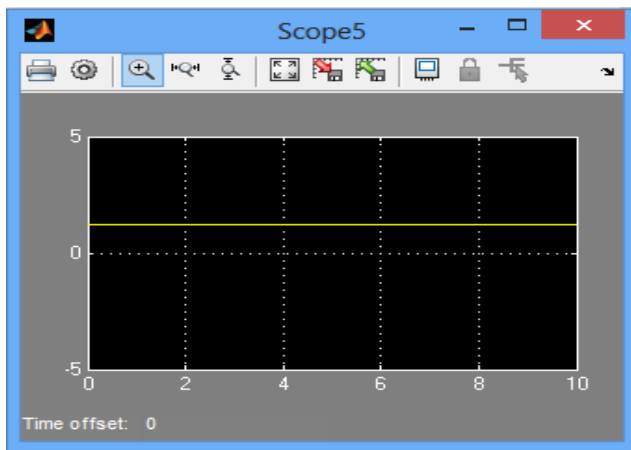


Gambar 5.5 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Kubus

Hasil tersebut menunjukkan bahwa selama proses simulasi berlangsung tidak terdapat perubahan yang relatif signifikan setelah proses PID. Jadi untuk skenario ini hasilnya cukup bagus.



Gambar 5.6 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Segilima

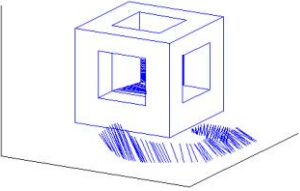


Gambar 5.7 Hasil Salah Satu *Block* Dari Proses PID Pada Objek Segitiga

5.4 Uji Coba Waktu Eksekusi Perangkat Lunak

Uji coba ini dilakukan dengan menghitung waktu eksekusi untuk setiap hasil akhir dari perangkat lunak dengan waktu simulasi 10 *seconds* dan berbagai *solver*. *Solver* yang digunakan untuk pengujian adalah ode23, ode45, dan ode113. Waktu eksekusi dihitung berdasarkan waktu yang dibutuhkan oleh simulink hingga selesai atau waktu simulasi sampai 10 *seconds*. Hasil yang diharapkan pada pengujian ini adalah sistem dapat merender objek dengan waktu eksekusi paling lama 4 jam.

Tabel 5.1 Hasil Uji Coba Waktu Eksekusi Objek Cube

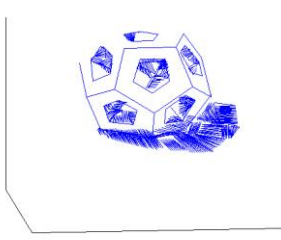
Objek			
			
Solver\Simulasi Ke-	I	II	III
Ode23	49 Menit	46,50 Menit	45,49 Menit
Ode45	100 Menit	99,20 Menit	100,18 Menit
Ode113	53,30 Menit	47,33 Menit	47,32 Menit

5.4.1 Uji Coba dan Evaluasi Waktu Eksekusi Perangkat Lunak

Uji coba dilakukan dengan menghitung *running time* menggunakan data objek 3D seperti yang disebutkan sebelumnya, dapat dilihat pada Gambar 5.1. Uji coba ini dilakukan dalam 3 kali proses simulasi untuk setiap objeknya.

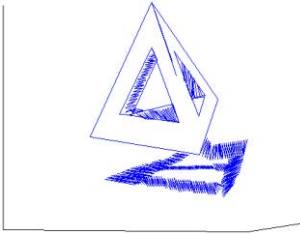
Berdasarkan uji coba yang telah dilakukan, nilai *running time* yang diperoleh untuk objek kubus akan ditunjukkan pada Tabel 5.1, untuk objek segilima akan ditunjukkan pada Tabel 5.2, untuk objek segitiga akan ditunjukkan pada Tabel 5.3. Untuk rata-rata *running time* dapat dilihat pada Tabel 5.4.

Tabel 5.2 Hasil Uji Coba Waktu Eksekusi Objek Segilima

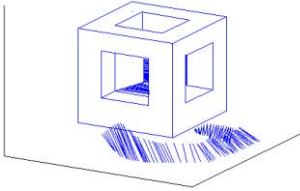
Objek			
			
Solver\Simulasi Ke-	I	II	III
Ode23	153,37 Menit	162,21 Menit	150,05 Menit
Ode45	278,17 Menit	263,26 Menit	263,09 Menit
Ode113	135 Menit	129,33 Menit	132,13 Menit

Hasil yang diperoleh menunjukkan ode45 menghasilkan performa yang kurang baik, sedangkan untuk ode23 dan ode113 menunjukkan performa yang cukup baik. Salah satu faktor yang mempengaruhi adalah tingkat akurasi yang menentukan, semakin tinggi tingkat akurasi suatu *solver*, performa yang dihasilkan akan semakin semakin kurang baik. Ode45 memiliki tingkat akurasi yang sedang, sedangkan ode23 tingkat akurasi yang rendah dan ode113 dengan tingkat akurasi dari rendah sampai tinggi, seperti yang dijelaskan pada subbab 2.1.


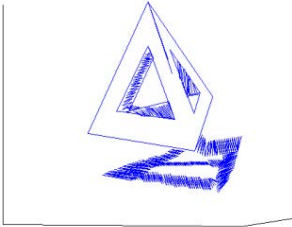
Tabel 5.3 Hasil Uji Coba Waktu Eksekusi Objek Segitiga

Objek			
			
Solver\Simulasi Ke-	I	II	III
Ode23	44,29 Menit	42,10 Menit	42,12 Menit
Ode45	86,06 Menit	84,18 Menit	76,16 Menit
Ode113	45,06 Menit	42,17 Menit	41,17 Menit

Tabel 5.4 Rata-rata Hasil Uji Coba Waktu Eksekusi (Bagian Pertama)

Objek	Solver	Rata-rata <i>Running Time</i>
	Ode23	46,9967 Menit
	Ode45	99,7933 Menit
	Ode113	49,3167 Menit

Tabel 5.5 Rata-rata Hasil Uji Coba Waktu Eksekusi (Bagian Kedua)

	Ode23	155,21 Menit
	Ode45	268,1733 Menit
	Ode113	132,1533 Menit
	Ode23	42,8367 Menit
	Ode45	82,1333 Menit
	Ode113	42,7867 Menit

LAMPIRAN A. Referensi Trajektori 3D

```
[-60, 22]
[-2 2]
[-2 2]
[-2 2]

1.25

[0.2 0.2 0.2]

[-1 -1 -1.25; -1 1 -1.25; -1 1 1.25; -1 -1 1.25;
-1 -1 -1.25; 1 -1 -1.25; 1 -1 1.25; -1 -1 1.25;
-1 1 1.25; 1 1 1.25; 1 -1 1.25; -1 -1 1.25;]

[0.5 0.5 1.25; -0.5 0.5 1.25; -0.5 -0.5 1.25;
0.5 -0.5 1.25; 0.5 0.5 1.25; -0.5 -0.075 1.25;]

[0.5 -1 -0.5; -0.5 -1 -0.5; -0.5 -1 0.75; 0.5 -
1 0.75; 0.5 -1 -0.5; 0.5 -0.42 -0.5; 0.5 -0.42
0.3; 0.5 -0.43 0.3; 0.5 -0.43 -0.5; 0.5 -0.445
0.3; 0.5 -0.445 -0.5; 0.5 -0.45 0.3; 0.5 -0.45
-0.5;]

[-1 0.5 -0.5; -1 0.5 0.75; -1 -0.5 0.75; -1 -
0.5 -0.5; -1 0.5 -0.5; -0.5 0.5 -0.5; -0.5 0.5
0.5; -0.5 0.5 -0.5; -0.5 -0.2 -0.5; -0.5 0.5 -
0.5; -0.15 0.1 -0.3; -0.15 0.1 0.3; -0.15 0.1 -
0.3; -0.15 0.1 -0.3; -0.2 0.12 -0.3; -0.27 -0.05
-0.3; -0.2 0.12 -0.3; -0.22 0.14 -0.325; -0.3 -
0.08 -0.31; -0.22 0.14 -0.325; -0.23 0.14 -0.33;
-0.315 -0.1 -0.33; -0.23 0.14 -0.33; -0.24 0.145
-0.335; -0.318 -0.11 -0.33; -0.24 0.145 -0.335;
-0.255 0.16 -0.345; -0.32 -0.1 -0.345; -0.255
0.16 -0.345; -0.26 0.165 -0.355; -0.325 -0.1 -
0.355; -0.26 0.165 -0.355; -0.27 0.225 -0.37; -
0.3325 -0.115 -0.37; -0.27 0.225 -0.37;
```

Gambar A 1. Cube.txt (Bagian Pertama)

```

-0.3 0.275 -0.38; -0.34 -0.12 -0.38; -0.3 0.275
-0.38; -0.35 0.275 -0.385; -0.345 -0.12 -0.385;
-0.35 0.275 -0.385; -0.355 0.28 -0.4; -0.35 -
0.125 -0.4; -0.355 0.28 -0.4; -0.36 0.325 -
0.415; -0.35 -0.13 -0.415; -0.36 0.325 -0.415;
-0.365 0.35 -0.435; -0.35 -0.13 -0.435; -0.365
0.35 -0.435; -0.375 0.38 -0.45; -0.35 -0.13 -
0.45; -0.375 0.38 -0.45; -0.375 0.45 -0.475; -
0.35 -0.13 -0.475; -0.375 0.45 -0.475; -0.375
0.5 -0.5; -0.35 -0.13 -0.5; -0.375 0.5 -0.5; -
0.5 0.5 -0.5; -0.15 0.1 -0.3; -0.15 0.1 -0.285;
-0.25 -0.07 -0.26; -0.15 0.1 -0.285; -0.15 0.1
-0.275; -0.25 -0.07 -0.245; -0.15 0.1 -0.275; -
0.15 0.1 -0.255; -0.25 -0.07 -0.225; -0.15 0.1
-0.255; -0.15 0.1 -0.24; -0.25 -0.07 -0.21; -
0.15 0.1 -0.24; -0.15 0.1 -0.22; -0.25 -0.07 -
0.19; -0.15 0.1 -0.22; -0.15 0.1 -0.21; -0.25 -
0.07 -0.18; -0.15 0.1 -0.21; -0.15 0.1 -0.19; -
0.25 -0.07 -0.16; -0.15 0.1 -0.19; -0.15 0.1 -
0.17; -0.25 -0.07 -0.14; -0.15 0.1 -0.17; -0.15
0.1 -0.16; -0.25 -0.07 -0.125; -0.15 0.1 -0.16;
-0.15 0.1 -0.14; -0.25 -0.07 -0.11; -0.15 0.1 -
0.14; -0.15 0.1 -0.12; -0.25 -0.07 -0.09; -0.15
0.1 -0.12; -0.15 0.1 -0.1; -0.25 -0.07 -0.07; -
0.15 0.1 -0.1; -0.15 0.1 -0.08; -0.25 -0.07 -
0.045; -0.15 0.1 -0.08; -0.15 0.1 -0.06; -0.25
-0.07 -0.03; -0.15 0.1 -0.06; -0.15 0.1 -0.05;
-0.25 -0.07 -0.02; -0.15 0.1 -0.05; -0.15 0.1 -
0.02; -0.25 -0.07 0.01; -0.15 0.1 -0.02; -0.15
0.1 0; -0.25 -0.07 0.03; -0.15 0.1 0; -0.15 0.1
0.025; -0.25 -0.07 0.055; -0.15 0.1 0.025; -0.15
0.1 0.05; -0.25 -0.07 0.08; -0.15 0.1 0.05; -
0.15 0.1 0.07; -0.25 -0.07 0.1; -0.15 0.1 0.07;
-0.15 0.1 0.1; -0.25 -0.07 0.13; -0.15 0.1 0.1;
-0.15 0.1 0.12; -0.25 -0.07 0.15; -0.15 0.1
0.12; -0.15 0.1 0.15; -0.25 -0.07 0.18; -0.15
0.1 0.15; -0.15 0.1 0.17; -0.25 -0.07 0.2; -0.15
0.1 0.17; -0.15 0.1 0.2; -0.25 -0.07 0.23; -0.15

```

Gambar A 2. Cube.txt (Bagian Kedua)

```

0.1 0.2; -0.15 0.1 0.22; -0.25 -0.07 0.25; -0.15
0.1 0.22; -0.15 0.1 0.25; -0.25 -0.07 0.28; -
0.15 0.1 0.25; -0.15 0.1 0.27; -0.25 -0.07 0.3;
-0.15 0.1 0.27; -0.15 0.1 0.3; -0.25 -0.07 0.33;
-0.15 0.1 0.3; -0.15 0.1 0.32; -0.25 -0.07 0.35;
-0.15 0.1 0.32; -0.15 0.1 -0.3;]

[-0.5 -1.2 -1.25; -0.3 -1.075 -1.25; -0.525 -
1.25 -1.25; -0.3 -1.1 -1.25; -0.525 -1.3 -1.25;
-0.25 -1.125 -1.25; -0.55 -1.325 -1.25; -0.2 -
1.1 -1.25; -0.575 -1.35 -1.25; -0.2 -1.15 -1.2;
-0.575 -1.4 -1.25; -0.1 -1.15 -1.2; -0.575 -1.45
-1.25; -0.05 -1.2 -1.175; -0.575 -1.5 -1.25; -
0.05 -1.275 -1.175; -0.625 -1.575 -1.25; -0.05
-1.35 -1.175; -0.715 -1.625 -1.25; -0.05 -1.4 -
1.175; -0.725 -1.7 -1.25; -0.05 -1.45 -1.175; -
0.735 -1.775 -1.25; -0.05 -1.5 -1.175; -0.745 -
1.825 -1.25; -0.05 -1.55 -1.175; -0.755 -1.875
-1.25; -0.05 -1.6 -1.175; -0.765 -1.95 -1.25; -
0.05 -1.65 -1.175; -0.775 -2.025 -1.25; -0.05 -
1.7 -1.175; -0.8 -2.1 -1.25; -0.05 -1.735 -
1.175;]

[-0.35 -2.05 -1; -0.4 -2.5 -1.25; -0.375 -2.05
-1; -0.45 -2.5 -1.25; -0.5 -2.05 -1; -0.6 -2.5
-1.25; -0.55 -2.05 -1; -0.675 -2.5 -1.25; -0.6
-2.05 -1; -0.75 -2.5 -1.25; -0.7 -2.05 -1; -0.9
-2.5 -1.25; -0.75 -2.05 -1; -0.975 -2.5 -1.25;
-0.8 -2.05 -1; -1.025 -2.5 -1.25; -0.9 -2.05 -
1; -1.175 -2.5 -1.25; -0.95 -2.05 -1; -1.25 -
2.5 -1.25; -1 -2.05 -1; -1.325 -2.5 -1.25; -1.15
-2.05 -1; -1.325 -2.5 -1.25; -1.2 -1.95 -1; -
1.4 -2.45 -1.25; -1.25 -1.95 -1; -1.475 -2.45 -
1.25; -1.3 -1.95 -1; -1.55 -2.45 -1.25;]

[-1.35 -1.95 -1; -1.625 -2.4 -1.25; -1.4 -1.95
-1; ]

```

Gambar A 3. Cube.txt (Bagian Ketiga)

```

[-1.825 -2.4 -1.25; -1.55 -1.95 -1; -1.9 -2.4 -
1.25;]

[-1.6 -1.95 -1; -1.975 -2.4 -1.25; -1.65 -1.9 -
1; -2.05 -2.4 -1.25;]

[-2.125 -2.25 -1.25; -1.7 -1.85 -1;]

[-2.2 -2.175 -1.25; -1.75 -1.8 -1; -2.2 -2.175
-1.25;]
[-2.275 -2.05 -1.25; -1.9 -1.75 -0.95;]
[-2.225 -1.5 -0.885; -2.4 -1.8 -1.25;]
[-2.4 -1.85 -1.25; -2.225 -1.575 -0.85;]
[-2.425 -1.7 -1.275; -2.235 -1.375 -0.865;]
[-2.475 -1.675 -1.25; -2.295 -1.375 -0.845;]
[-2.5 -1.65 -1.25; -2.325 -1.35 -0.815;]
[-2.55 -1.575 -1.25; -2.4 -1.3 -0.775;]
[-2.575 -1.525 -1.25; -2.475 -1.25 -0.725; ]
[-2.6 -1.45 -1.25; -2.525 -1.15 -0.675; -2.625
-1.425 -1.25; -2.55 -1.1 -0.65;]
[-2.625 -1.375 -1.25; -2.6 -1.075 -0.575;]
[-2.65 -1.375 -1.25; -2.6 -1.05 -0.5; -2.725 -
1.395 -1.2; -2.65 -1.025 -0.45;]
[-2.65 -1.225 -1.25; -2.75 -0.975 -0.3; -2.675
-1.2 -1.25; -2.775 -0.925 -0.275;]
[-2.7 -1.175 -1.225; -2.8 -0.9 -0.275; -2.725 -
1.15 -1.215; -2.825 -0.885 -0.275;]
[-2.75 -1.1 -1.2; -2.85 -0.85 -0.3; -2.775 -
1.075 -1.185; -2.875 -0.825 -0.325; -2.8 -1.05
-1.165; -2.9 -0.8 -0.35;]
[-2.825 -1.0 -1.125; -2.925 -0.775 -0.375;]
[-2.85 -0.975 -1.095; -2.95 -0.75 -0.415; -2.875
-0.95 -1.05; -2.975 -0.725 -0.45;]
[-2.9 -0.9 -0.975; -3 -0.7 -0.5; -2.925 -0.875
-0.95; -3.025 -0.675 -0.525;]

```

Gambar A 4. Cube.txt (Bagian Keempat)


```

[-6, 14]
[-2 2]
[-2 2]
[-2 1]

1.5

[0.1 0.1 0.1]

[-0.425 -1 1; 0.4 -1 1; 0.65 0.25 1; 0.95 0.35
0.55; 1 -0.65 0.05; 0.615 -1.5 0.4; 0.4 -1 1;
0.615 -1.5 0.4; -0.025 -1.75 0; -0.62 -1.35
0.35; -0.425 -1 1; -0.62 -1.35 0.35; -1.015 -
0.85 0.125; -1 0.35 0.5; -1 0.35 0.5; -1.015 -
0.85 0.125; -0.75 -0.5 -0.5; 0 -1.25 -0.5; -
0.025 -1.75 0; 0 -1.25 -0.5; 0.6 -0.6 -0.5; 1 -
0.65 0.1;]

[0 0.5 1; 0.25 0.25 1; 0.1 -0.35 1; -0.25 -0.35
1; -0.3 0.25 1; 0 0.5 1; -0.1 0.465 1; -0.2 -
0.35 1; -0.125 0.445 1; -0.215 -0.35 1; -0.15
0.425 1; -0.225 -0.35 1; -0.175 0.4 1; -0.235 -
0.35 1; -0.2 0.375 1; -0.245 -0.35 1; -0.225
0.35 1; -0.25 -0.35 1; -0.25 0.325 1; -0.275 -
0.3 1; -0.275 0.35 1; -0.3 -0.25 1;]

[0.65 0 0.85; 0.9 0.25 0.5; 0.9 -0.25 0.35;
0.675 -0.75 0.5; 0.5 -0.75 0.85; 0.65 0 0.85;
0.9 0.25 0.5; 0.8 0.1 0.5; 0.9 0.25 0.475; 0.85
0.25 0.575; 0.85 0.1 0.49; 0.825 0.1 0.5; 0.85
0.25 0.575; 0.85 0.3 0.575; 0.825 0.15 0.5; 0.8
0.15 0.5; 0.8 0.025 0.6; 0.8 0.075 0.6; 0.775
0.15 0.525; 0.75 0.15 0.55; 0.8 0.1 0.625; 0.8
0.125 0.635; 0.735 0.15 0.55; 0.7 0.15 0.6; 0.8
0.15 0.65; 0.785 0.2 0.66; 0.7 0.15 0.6; 0.675
0.15 0.625; 0.78 0.25 0.65; 0.78 0.25 0.675;
0.665 0.15 0.625; 0.65 0.15 0.65; 0.76 0.25
0.675; 0.725 0.25 0.7; 0.64 0.15 0.65;

```

Gambar A 5. Segilima.txt (Bagian Pertama)

```
0.625 0.15 0.675; 0.75 0.25 0.715; 0.735 0.25
0.725; 0.625 0.15 0.7; 0.7 0.25 0.75; 0.6 0.15
0.725; 0.675 0.15 0.8; 0.65 0 0.85; 0.5 -0.75
0.85; 0.625 -0.75 0.6; 0.7 -0.75 0.635; 0.625 -
0.75 0.6; 0.625 -0.75 0.575; 0.75 -0.75 0.625;
0.625 -0.75 0.575; 0.65 -0.75 0.55; 0.7 -0.75
0.6; 0.65 -0.75 0.55; 0.65 -0.75 0.525; 0.7 -
0.75 0.575; 0.65 -0.75 0.525; 0.675 -0.75 0.5;
0.75 -0.75 0.575; 0.675 -0.75 0.5; 0.7 -0.75
0.475; 0.75 -0.75 0.55; 0.7 -0.75 0.475;]
```

```
[0.175 -1.25 0.84; -0.2 -1.25 0.85; -0.3 -1.2
0.45; 0.025 -1 0.2; 0.275 -1.2 0.45; 0.175 -1.25
0.84; 0.1 -1.275 0.8; -0.2 -1.275 0.8; -0.15 -
1.275 0.8; -0.185 -1.25 0.85; -0.15 -1.275 0.8;
-0.125 -1.275 0.8; -0.165 -1.25 0.85; -0.125 -
1.275 0.8; -0.1 -1.275 0.8; -0.145 -1.25 0.85;
-0.1 -1.275 0.8; -0.085 -1.275 0.8; -0.125 -1.25
0.85; -0.085 -1.275 0.8; -0.055 -1.275 0.8; -
0.1 -1.25 0.85; -0.055 -1.275 0.8; -0.025 -1.275
0.8; -0.085 -1.25 0.85; -0.025 -1.275 0.8; 0.015
-1.275 0.8; -0.05 -1.25 0.85; 0.015 -1.275 0.8;
0.035 -1.275 0.8; -0.025 -1.25 0.85; 0.035 -
1.275 0.8; 0.085 -1.275 0.8; 0 -1.25 0.85; 0.085
-1.275 0.8; 0.115 -1.275 0.8; 0.015 -1.25 0.85;
0.115 -1.275 0.8; 0.115 -1.275 0.8; 0.2 -1.2
0.5; 0.115 -1.275 0.8; 0.085 -1 0.75; 0.2 -1.275
0.85; 0.095 -1.05 0.735; 0.2 -1.275 0.825; 0.115
-1.075 0.715; 0.215 -1.3 0.8; 0.095 -1.2 0.7;
0.215 -1.3 0.78; 0.08 -1.15 0.65; 0.215 -1.3
0.75; 0.1 -1.2 0.65; 0.235 -1.3 0.7; 0.185 -1.25
0.85; 0.125 -1.3 0.8; 0.2 -1.275 0.84; 0.115 -
1.3 0.75; 0.2 -1.3 0.825; 0.135 -1.325 0.725;
0.2 -1.3 0.825; 0.15 -1.35 0.725; 0.225 -1.325
0.8; 0.15 -1.35 0.685; 0.235 -1.325 0.775; 0.165
-1.35 0.65; 0.235 -1.325 0.775; 0.165 -1.1
0.625; 0.225 -1.5 0.75; 0.15 -1.5 0.625; 0.185
-1.85 0.75; 0.15 -1.65 0.6; 0.2 -1.85 0.7;
```

Gambar A 6. Segilima.txt (Bagian Kedua)

```

0.175 -1.75 0.6; 0.275 -1.2 0.55; 0.2 -1.2 0.5;
0.185 -1.1 0.5; 0.15 -1.1 0.6; 0.185 -1.1 0.5;
0.165 -1 0.5; 0.15 -1 0.55; 0.165 -1 0.5; 0.135
-0.95 0.5; 0.135 -0.95 0.525; 0.135 -0.95 0.5;
0.1 -0.85 0.5; 0.15 -1.35 0.8; 0.1 -0.85 0.5; -
0.235 -1.275 0.75; -0.235 -1.275 0.75; -0.2 -
1.275 0.8; -0.2 -1.3 0.75; -0.2 -1.275 0.8; -
0.185 -1.275 0.8; -0.185 -1.3 0.725; -0.185 -
1.275 0.8; -0.165 -1.275 0.8; -0.15 -1.3 0.7; -
0.165 -1.275 0.8; -0.155 -1.275 0.8; -0.115 -
1.35 0.675; -0.155 -1.275 0.8; -0.135 -1.275
0.8; -0.075 -1.3 0.65; -0.135 -1.275 0.8; -0.115
-1.275 0.8; -0.035 -1.3 0.625; -0.115 -1.275
0.8; -0.095 -1.275 0.8; -0.015 -1.3 0.625; -
0.095 -1.275 0.8; -0.065 -1.275 0.8; 0.015 -1.3
0.625; -0.065 -1.275 0.8; -0.035 -1.275 0.8;
0.03 -1.3 0.625; -0.035 -1.275 0.8; -0.2 -1.275
0.8; -0.235 -1.275 0.75; 0.1 -0.85 0.5; 0.05 -
1.1 0.565; 0.15 -1.35 0.8; 0.05 -1.1 0.565; 0.1
-0.85 0.5; -0.025 -0.9 0.275; 0.1 -0.85 0.5; 0.1
-0.85 0.5; 0.085 -0.85 0.475; 0.15 -0.85 0.475;
0.085 -0.85 0.475; 0.065 -0.85 0.465; 0.2 -0.85
0.465; 0.065 -0.85 0.465; 0.05 -1 0.46; 0.2 -
1.1 0.46; 0.05 -1 0.46; 0.025 -1.05 0.45; 0.18
-1.05 0.45; 0.025 -1.05 0.45; 0.0 -1.1 0.44;
0.135 -1.1 0.44; 0.0 -1.1 0.44; -0.015 -1.2
0.43; 0.125 -1.2 0.43; -0.015 -1.2 0.43; -0.035
-1.3 0.42; 0.085 -1.3 0.42; -0.035 -1.3 0.42; -
0.085 -1.4 0.41; 0.05 -1.4 0.41; -0.085 -1.4
0.41; -0.1 -1.5 0.4; 0.025 -1.5 0.4; -0.1 -1.5
0.4; -0.025 -0.9 0.275; 0.1 -0.85 0.5; 0.025 -
0.85 0.45; -0.075 -1 0.3; 0.025 -0.85 0.45; -
0.25 -1.275 0.7; -0.235 -1.275 0.75; -0.235 -
1.275 0.75; -0.185 -1.275 0.65; -0.2 -1.275
0.725; -0.15 -1.275 0.625; -0.185 -1.2 0.725; -
0.1 -1.275 0.6; -0.15 -1.5 0.725; -0.05 -1.275
0.525; -0.1 -1.5 0.7; 0 -1.275 0.525;]

```

Gambar A 7. Segilima.txt (Bagian Ketiga)

```

[-0.7 0 0.85; -0.6 -0.85 0.85; -0.65 -1 0.55; -
0.85 -0.25 0.3; -0.85 0 0.6; -0.7 0 0.85; -0.85
0 0.6; -0.55 0 0.45; -0.85 0 0.6; -0.65 0 0.525;
-0.85 0 0.6; -0.85 0 0.625; -0.65 0 0.535; -0.85
0 0.625; -0.85 -0.1 0.675; -0.675 0.0 0.575; -
0.85 -0.1 0.675; -0.85 -0.35 0.725; -0.675 0.0
0.6; -0.85 -0.35 0.725; -0.825 -0.35 0.75; -
0.675 0.0 0.625; -0.825 -0.35 0.75; -0.625 0.0
0.665; -0.825 -0.35 0.75; -0.62 0.0 0.7; -0.825
-0.35 0.75; -0.62 0.0 0.725; -0.825 -0.35 0.75;
-0.615 0.0 0.765; -0.825 -0.35 0.75; -0.615 0.0
0.785; -0.825 -0.35 0.75; -0.825 -0.4 0.775; -
0.615 0.0 0.8; -0.825 -0.4 0.775; -0.825 -0.45
0.8; -0.635 0.0 0.825; -0.825 -0.45 0.8; -0.825
-0.5 0.825; -0.65 0.0 0.85; -0.825 -0.5 0.825;
-0.85 0 0.6; -0.7 0 0.85; -0.615 0.0 0.8; -0.6
-0.85 0.85; -0.615 0.0 0.8; -0.615 0.0 0.775; -
0.6 -0.85 0.825; -0.615 0.0 0.775; -0.615 0.0
0.75; -0.6 -0.85 0.81; -0.615 0.0 0.75; -0.615
0.0 0.725; -0.6 -0.85 0.785; -0.615 0.0 0.725;
-0.615 0.0 0.71; -0.6 -0.85 0.765; -0.615 0.0
0.71; -0.615 0.0 0.685; -0.6 -0.85 0.745; -0.615
0.0 0.685; -0.615 0.0 0.65; -0.6 -0.85 0.71; -
0.65 -1 0.55; -0.55 0 0.45; -0.55 0.1 0.45; -
0.625 -0.85 0.71; -0.55 0.1 0.45; -0.55 0.1
0.375; -0.55 0.1 0.45; -0.575 0.2 0.45; -0.625
-0.85 0.71; -0.575 0.2 0.45; -0.55 0.2 0.35; -
0.575 0.2 0.45; -0.6 0.25 0.45; -0.635 -0.85
0.75; -0.6 0.25 0.45; -0.575 0.25 0.35; -0.6
0.25 0.45; -0.6 0.35 0.45; -0.65 -0.85 0.75; -
0.6 0.35 0.45; -0.6 0.35 0.375; -0.6 0.35 0.45;
-0.625 0.4 0.45; -0.675 -0.85 0.75; -0.625 0.4
0.45; -0.6 0.4 0.385; -0.625 0.4 0.45; -0.65
0.45 0.45; -0.7 -0.85 0.75; -0.65 0.45 0.45; -
0.625 0.45 0.4; -0.65 0.45 0.45; -0.675 0.5
0.45; -0.7 -0.85 0.775; -0.675 0.5 0.45;

```

Gambar A 8. Segilima.txt (Bagian Keempat)

```

-0.685 0.55 0.45; -0.725 -0.85 0.775; -0.685
0.55 0.45;]

[-0.65 -0.9 -0.3; -0.3 -1.2 -0.3; -0.25 -1.25 -
0.05; -0.55 -1.3 0.15; -0.75 -1 0; -0.65 -0.9 -
0.3; -0.65 -0.9 -0.3; -0.45 -1.3 0.1; -0.55 -
1.3 0.15; -0.45 -1.3 0.1; -0.45 -1.3 0.1; -0.575
-1.3 0.125; -0.45 -1.3 0.1; -0.4 -1 0.05; -0.575
-1.3 0.125; -0.4 -1 0.05; -0.35 -0.7 0.0; -0.575
-1.3 0.125; -0.4 -0.7 0.0; -0.4 -0.7 -0.025; -
0.575 -1.3 0.1; -0.4 -0.7 -0.025; -0.4 -0.5 -
0.075; -0.575 -1.3 0.085; -0.4 -0.5 -0.075; -
0.4 -0.4 -0.115; -0.6 -1.1 0.05; -0.4 -0.4 -
0.115; -0.4 -0.3 -0.15; -0.6 -1.1 0.025; -0.4 -
0.3 -0.15; -0.4 -0.2 -0.175; -0.6 -1.1 0.0; -
0.4 -0.2 -0.175; -0.4 -0.075 -0.215; -0.6 -0.95
-0.025; -0.4 -0.075 -0.215; -0.4 0.0 -0.25; -
0.6 -0.95 -0.05; -0.4 0.0 -0.25; -0.6 -0.7 -0.1;
-0.4 0.0 -0.25; -0.65 -0.8 -0.1; -0.4 0.0 -0.25;
-0.675 -1 -0.1; -0.4 0.0 -0.25; -0.7 -1.15 -0.1;
-0.4 0.0 -0.25; -0.71 -1.2 -0.1; -0.4 0.0 -0.25;
-0.75 -1.35 -0.1; -0.4 0.0 -0.25; -0.4 0.2 -0.3;
-0.725 -1.35 -0.115; -0.4 0.2 -0.3; -0.725 -1.35
-0.15; -0.4 0.2 -0.3; -0.7 -1.35 -0.175; -0.4
0.2 -0.3; -0.7 -1.35 -0.2; -0.4 0.2 -0.3; -0.65
-0.9 -0.3; -0.75 -1 0; -0.75 -1 0; -0.7 -1 -0.1;
-0.75 -1 0; -0.745 -1.15 0.025; -0.7 -1.15 -0.1;
-0.745 -1.15 0.025; -0.735 -1.165 0.05; -0.65 -
0.8 -0.1; -0.735 -1.165 0.05; -0.75 -1.65 0.1;
-0.625 -0.85 -0.1; -0.75 -1.65 0.1; -0.75 -1.65
0.125; -0.565 -0.5 -0.15; -0.75 -1.65 0.125; -
0.75 -1.75 0.15; -0.6 -0.95 -0.075; -0.75 -1.75
0.15; -0.725 -1.8 0.175; -0.75 -1 0; -0.65 -0.9 -0.3;
-0.4 0.2 -0.3; -0.4 0.4 -0.4; -0.38 0.2 -0.3; -
0.375 0.45 -0.4; -0.365 0.2 -0.3; -0.35 0.475 -
0.4; -0.335 0.2 -0.3; -0.325 0.48 -0.4; -0.335
0.2 -0.3; -0.4 0.2 -0.3; -0.65 -0.9 -0.3;

```

Gambar A 9. Segilima.txt (Bagian Kelima)

```

-0.575 -0.8 -0.325; -0.5 -0.2 -0.325; -0.55 -
0.8 -0.325; -0.475 -0.2 -0.325; -0.525 -0.8 -
0.325; -0.45 -0.2 -0.325; -0.5 -0.8 -0.325; -
0.425 -0.2 -0.325; -0.475 -0.8 -0.325; -0.4 -
0.2 -0.325; -0.45 -0.8 -0.325; -0.375 -0.2 -
0.25; -0.425 -0.8 -0.325; -0.35 -0.2 -0.25; -
0.4 -0.8 -0.325; -0.325 -0.2 -0.25; -0.375 -0.8
-0.325; -0.3 -0.2 -0.25; -0.35 -0.8 -0.325; -
0.275 -0.2 -0.25; -0.325 -0.8 -0.325; -0.275 -
0.2 -0.25; -0.3 -0.8 -0.325; -0.3 -1.2 -0.3; -
0.3 -0.8 -0.325; -0.3 -1.2 -0.275; -0.3 -0.8 -
0.3; -0.3 -1.2 -0.25; -0.3 -0.8 -0.275; -0.275
-1.2 -0.225; -0.325 -0.8 -0.225; -0.275 -1.2 -
0.175; -0.325 -0.8 -0.2; -0.275 -1.2 -0.15; -
0.325 -0.8 -0.175; -0.275 -1.2 -0.125; -0.325 -
0.8 -0.15; -0.25 -1.2 -0.085;]

```

```

[ 0.2 -1.15 -0.3; 0.5 -1 -0.3; 0.8 -0.1 -0.1;
0.5 -1.5 0.25; 0.25 -0.9 -0.1; 0.2 -1.15 -0.3;
0.25 -0.9 -0.1; 0.5 -1.5 0.25; 0.8 -0.1 -0.1;
0.8 -0.1 -0.1; 0.5 -1.35 0.2; 0.8 -0.1 -0.1; 0.5
-1.25 0.175; 0.8 -0.1 -0.1; 0.465 -1.1 0.15; 0.8
-0.1 -0.1; 0.475 -0.8 0.085; 0.8 -0.1 -0.1;
0.475 -0.8 0.05; 0.8 -0.1 -0.1; 0.475 -0.8
0.025; 0.8 -0.1 -0.1; 0.475 -0.8 -0.015; 0.8 -
0.1 -0.1; 0.8 0.05 -0.15; 0.475 -0.8 -0.035; 0.8
0.05 -0.15; 0.8 0.075 -0.175; 0.475 -0.8 -0.035;
0.8 0.075 -0.175; 0.8 0.15 -0.2; 0.475 -0.8 -
0.075; 0.8 0.15 -0.2; 0.8 0.225 -0.225; 0.475 -
0.825 -0.1; 0.8 0.225 -0.225; 0.8 0.275 -0.25;
0.475 -0.8 -0.135; 0.8 0.275 -0.25; 0.8 0.375 -
0.275; 0.475 -0.785 -0.165; 0.8 0.375 -0.275;
0.8 0.45 -0.3; 0.475 -0.785 -0.2; 0.8 0.45 -0.3;
0.8 0.525 -0.325; 0.5 -0.785 -0.185; 0.8 0.525
-0.325; 0.465 -1.1 0.15; 0.8 0.525 -0.325; 0.8
0.75 -0.375; 0.525 -0.785 -0.2; 0.8 0.75 -0.375;
0.475 -0.8 0.085; 0.8 0.75 -0.375; 0.8 0.8 -0.4;
0.525 -0.785 -0.225; 0.8 0.8 -0.4;

```

Gambar A 10. Segilima.txt (Bagian Keenam)

```

0.475 -0.8 0.05; 0.8 0.8 -0.4; 0.8 1 -0.425;
0.525 -0.785 -0.25; 0.8 1 -0.425; 0.475 -0.8
0.025; 0.8 1 -0.425; 0.5 -1 -0.3; 0.35 -0.7 -
0.1; 0.5 -1 -0.3; 0.475 -0.9 -0.25; 0.35 -1 -
0.2; 0.475 -1 -0.275; 0.3 -1.1 -0.2; 0.5 -0.8 -
0.325; 0.25 -1.25 -0.2; 0.45 -0.7 -0.35; 0.2 -
1.5 -0.2; 0.4 -0.6 -0.35; 0.175 -1.5 -0.225;]

[-0.5 -0.75 -0.5; -0.1 -1.3 -0.5; 0 -1.275 -0.5;
0.6 -0.6 -0.5; 0 -1.285 -0.5; -0.1 -1.325 -0.5;
-0.5 -0.85 -0.5; -0.1 -1.345 -0.5; 0 -1.275 -
0.5; 0.6 -0.7 -0.5; 0 -1.295 -0.5; -0.1 -1.365
-0.5; -0.75 -0.5 -0.5; -0.5 -0.95 -0.515; -0.1
-1.39 -0.515; 0 -1.275 -0.515; 0.6 -0.8 -0.515;
0 -1.305 -0.515; -0.1 -1.405 -0.515;]

[-0.8 -0.5 -0.5; -0.8 -1 -0.5; -0.775 -0.4 -0.5;
-0.775 -1 -0.5; -0.75 -0.35 -0.5; -0.735 -1.05
-0.5; -0.7 -0.3 -0.55; -0.685 -1.1 -0.55; -0.65
-0.25 -0.55; -0.635 -1.15 -0.55; -0.625 -0.225
-0.55; -0.615 -1.175 -0.55; -0.6 -0.2 -0.565; -
0.585 -1.2 -0.565; -0.575 -0.175 -0.565; -0.565
-1.225 -0.565; -0.55 -0.15 -0.565; -0.545 -1.25
-0.565; -0.525 -0.125 -0.575; -0.515 -1.275 -
0.575; -0.5 -0.1 -0.575; -0.485 -1.3 -0.575; -
0.475 -0.075 -0.575; -0.465 -1.325 -0.575; -0.45
-0.125 -0.585; -0.445 -1.35 -0.585; -0.425 -
0.125 -0.585; -0.415 -1.375 -0.585; -0.4 -0.125
-0.585; -0.385 -1.4 -0.585; -0.375 -0.125 -
0.595; -0.365 -1.425 -0.595; -0.35 -0.125 -
0.595; -0.345 -1.45 -0.595; -0.325 -0.125 -
0.595; -0.315 -1.475 -0.595; -0.3 -0.1 -0.605;
-0.285 -1.5 -0.605; -0.275 -0.075 -0.605; -0.265
-1.525 -0.605; -0.25 -0.05 -0.605; -0.245 -1.55
-0.605; -0.225 -0.025 -0.615; -0.215 -1.575 -
0.615; -0.2 0.0 -0.615; -0.185 -1.6 -0.615; -
0.175 0.025 -0.615; -0.165 -1.625 -0.615; -0.15
0.05 -0.625; -0.145 -1.65 -0.625;

```

Gambar A 11. Segilima.txt (Bagian Ketujuh)

```
-0.125 0.075 -0.625; -0.115 -1.675 -0.625; -0.1
0.1 -0.625; -0.085 -1.6 -0.625; -0.075 0.125 -
0.635; -0.065 -1.6 -0.635; -0.05 0.135 -0.635;
-0.045 -1.7 -0.635; -0.025 0.135 -0.635; -0.015
-1.6 -0.635; 0.0 0.135 -0.645; 0.015 -1.7 -
0.645; 0.025 0.135 -0.645; 0.045 -1.6 -0.645;
0.05 0.115 -0.645; 0.065 -1.5 -0.645; 0.075 0.1
-0.655; 0.085 -1.475 -0.655; 0.1 0 -0.655; 0.115
-1.475 -0.655; 0.125 -0.025 -0.655; 0.145 -1.475
-0.655;]
```

```
[ 0.175 -0.75 -0.655; 0.235 -1.75 -0.655; 0.2 -
0.75 -0.655; 0.265 -1.75 -0.655; 0.225 -0.75 -
0.655; 0.295 -1.75 -0.655; 0.25 -0.75 -0.655;
0.335 -1.75 -0.655; 0.275 -0.75 -0.655; 0.365 -
1.75 -0.655; 0.3 -0.75 -0.655; 0.395 -1.75 -
0.655; 0.325 -0.75 -0.655; 0.435 -1.75 -0.655;
0.35 -0.75 -0.655; 0.465 -1.75 -0.655; 0.375 -
0.75 -0.655; 0.495 -1.75 -0.655; 0.4 -0.1 -
0.655; 0.535 -1.95 -0.655; 0.425 -0.1 -0.655;
0.565 -1.95 -0.655; 0.45 -0.1 -0.655; 0.565 -
1.95 -0.655; 0.45 -0.1 -0.655; 0.595 -1.95 -
0.655; 0.45 -0.1 -0.655; 0.635 -1.95 -0.655;
0.45 -0.1 -0.655; 0.665 -1.95 -0.655; 0.475 -
0.1 -0.655; 0.695 -1.95 -0.655; 0.475 -0.1 -
0.655; 0.735 -1.95 -0.655; 0.5 -0.1 -0.655;
0.765 -1.95 -0.655; 0.525 -0.1 -0.655; 0.795 -
1.95 -0.655; 0.55 -0.1 -0.655; 0.835 -1.95 -
0.655; 0.575 -0.1 -0.655; 0.865 -1.95 -0.655;
0.6 -0.1 -0.655; 0.895 -1.95 -0.655; 0.6 -0.1 -
0.655; 0.9 -1.75 -0.655; 0.6 -0.1 -0.655; 0.935
-1.65 -0.655; 0.6 -0.1 -0.655; 0.965 -1.5 -
0.655; 0.6 -0.1 -0.655; 0.995 -1.5 -0.655; 0.75
-0.5 -0.655; 1.035 -1.35 -0.655; 0.775 -0.425 -
0.655; 1.065 -1.35 -0.655; 0.775 -0.375 -0.655;
1.095 -1.2 -0.655; 0.775 -0.375 -0.655; 1.135 -
1.05 -0.655; 0.775 -0.325 -0.655; 1.165 -0.95 -
0.655;
```

Gambar A 12. Segilima.txt (Bagian Kedelapan)


```
0.8 -0.3 -0.655; 1.195 -0.85 -0.655; 0.825 -0.2
-0.645; 1.215 -0.6 -0.645; 0.85 -0.15 -0.635;
1.235 -0.45 -0.635; 0.875 -0.1 -0.625; 1.265 -
0.3 -0.625; 0.9 0 -0.615; 1.295 -0.15 -0.615;
0.925 0.1 -0.595; 1.315 0 -0.595; 0.975 0.2 -
0.595; 1.35 0.1 -0.595; 0.95 0.25 -0.595; 1.25
0.175 -0.595;]
```

```
[0.925 0.7 -0.595; 0.925 0.8 -0.585; 0.975 0.7
-0.595; 0.95 0.9 -0.585; 1.025 0.5 -0.595; 0.975
1 -0.585; 1.075 0.475 -0.595; 1 1.15 -0.585;
1.075 0.475 -0.595; 1.05 1.175 -0.585; 1.075
0.475 -0.595; 1.075 1.175 -0.585; 1.1 0.425 -
0.595; 1.1 1.175 -0.585; 1.125 0.3 -0.595; 1.125
1.195 -0.585; 1.15 0.25 -0.595; 1.15 1.5 -0.585;
1.175 0.225 -0.595; 1.175 1.6 -0.585; 1.2 0.2 -
0.595; 1.2 1.6 -0.55; 1.225 0.175 -0.595; 1.225
1.6 -0.535; 1.25 0.3 -0.595; 1.25 1.6 -0.515;
1.25 0.35 -0.595; 1.275 1.6 -0.485; 1.275 1.6 -
0.485; 1.1 1.175 -0.585; 1.25 1.675 -0.485;
1.075 1.215 -0.585; 1.225 1.695 -0.485; 1.05
1.235 -0.585;]
```

```
[1.275 1.55 -0.465; 1.6 1.5 -0.5; 1.275 1.45 -
0.515; 1.6 1.45 -0.515; 1.275 1.425 -0.53; 1.6
1.425 -0.53; 1.275 1.4 -0.545; 1.6 1.4 -0.545;
1.275 1.375 -0.56; 1.6 1.375 -0.56; 1.275 1.35
-0.575; 1.6 1.35 -0.575; 1.275 1.325 -0.59; 1.6
1.325 -0.59; 1.275 1.3 -0.61; 1.6 1.3 -0.61;
1.275 1.275 -0.61; 1.6 1.3 -0.61; 1.275 1.25 -
0.635; 1.6 1.25 -0.635; 1.275 1.225 -0.66; 1.6
1.225 -0.665; 1.2 1.2 -0.68; 1.6 1.2 -0.68; 1.2
1.175 -0.695; 1.6 1.175 -0.695; 1.2 1.15 -0.715;
1.6 1.15 -0.715; 1.175 1.125 -0.73; 1.6 1.125 -
0.73; 1.175 1.1 -0.745; 1.6 1.1 -0.745; 1.6 1.1
-0.745; 1.45 1.075 -0.76; 1.6 1.1 -0.785; 1.5
1.1 -0.795; 1.625 1.1 -0.825; 1.5 1.1 -0.795;
1.5 1.1 -0.795; 1.5 1.1 -1; 1.625 1.1 -0.8;
```

Gambar A 13. Segilima.txt (Bagian Kesembilan)

```
1.5 1.1 -1; 1.6 1.1 -0.8; 1.5 1.1 -1; 1.575 1.1
-0.8; 1.5 1.1 -1; 1.55 1.1 -0.8; 1.5 1.1 -1;
1.525 1.1 -0.8; 1.5 1.1 -1; 1.5 1.1 -0.8; 1.475
1.1 -1; 1.475 1.1 -0.8; 1.45 1.1 -1; 1.45 1.1 -
0.775; 1.425 1.1 -1; 1.425 1.1 -0.775; 1.4 1.1
-1; 1.4 1.1 -0.775; 1.375 1.1 -1; 1.375 1.1 -
0.775; 1.35 1.1 -1; 1.35 1.1 -0.775;]
```

```
[1 1.75 -1.1; 1 0.9 -0.675; 0.975 1.75 -1.1;
0.975 0.9 -0.675; 0.95 1.75 -1.1; 0.95 0.9 -
0.675; 0.925 1.75 -1.1; 0.925 0.9 -0.675; 0.9
1.75 -1.1; 0.9 0.9 -0.675; 0.875 1.75 -1.1;
0.875 0.9 -0.675; 0.85 1.75 -1.1; 0.85 0.9 -
0.675; 0.825 0.9 -0.675; 0.9 0.9 -0.575; 0.9 0.9
-0.575; 0.8 0.9 -0.55; 0.9 0.9 -0.575; 0.875 0.9
-0.6; 0.775 0.9 -0.565; 0.875 0.9 -0.6; 0.85 0.9
-0.625; 0.775 0.9 -0.6; 0.85 0.9 -0.625; 0.825
0.9 -0.65; 0.775 0.9 -0.625; 0.825 0.9 -0.65;
0.825 0.9 -0.675; 0.75 0.9 -0.635; 0.825 0.9 -
0.675; 0.81 0.9 -0.69; 0.71 0.9 -0.65; 0.81 0.9
-0.69; 0.795 0.9 -0.715; 0.685 0.9 -0.675; 0.795
0.9 -0.715; 0.78 0.9 -0.73; 0.65 0.9 -0.69; 0.78
0.9 -0.73; 0.765 0.9 -0.745; 0.6 0.9 -0.7; 0.765
0.9 -0.745; 0.74 0.9 -0.76; 0.55 0.9 -0.71; 0.74
0.9 -0.76; 0.725 0.9 -0.775; 0.5 0.9 -0.71;
0.725 0.9 -0.775; 0.71 0.9 -0.79; 0.485 0.9 -
0.71; 0.71 0.9 -0.79; 0.685 0.9 -0.805; 0.45 0.9
-0.715; 0.685 0.9 -0.805; 0.67 0.9 -0.83; 0.425
0.9 -0.735; 0.67 0.9 -0.83; 0.655 0.9 -0.845;
0.375 0.9 -0.76; 0.655 0.9 -0.845; 0.64 0.9 -
0.86; 0.325 0.9 -0.775; 0.64 0.9 -0.86; 0.625
0.9 -0.885; 0.275 0.9 -0.795; 0.625 0.9 -0.885;
0.6 0.9 -0.9; 0.225 0.9 -0.815; 0.6 0.9 -0.9;]
```

Gambar A 14. Segilima.txt (Bagian Kesepuluh)

```

[-78, 2]
[-2 2]
[-2 2]
[-2 1]

1.25

[0.2 0.2 0.2]

[0 0 1; 0 1 -0.8; 0 0 1; 0.4 -1 -0.3; -1 -1 -1;
0 1 -0.8; 0 0 1; -1 -0.55 0.25; 0 0 1; -1 -0.625
0.05; ]

[0 0 0.4; -0.1 0.5 -0.5; -0.1 0.3 -0.5; -0.4 -
0.175 0.25; 0 0 0.4; -0.2 -0.5 -0.6; -0.4 0.3 -
0.5; -0.1 -0.375 -0.35; -0.2 -0.5 -0.6;]

[-0.1 -0.4 0.35; -0.6 -0.9 -0.5; 0.6 -0.75 -
0.25; -0.1 -0.4 0.35; 0.6 -0.75 -0.25; -0.1 -
0.675 -0.275;]

[-0.2 -0.5 -0.6; -0.1 -0.35 -0.35; -0.2 -0.5 -
0.6; -0.2 -0.45 -0.575; -0.1 -0.35 -0.4; -0.2 -
0.45 -0.575; -0.2 -0.425 -0.575; -0.1 -0.325 -
0.4; -0.2 -0.425 -0.575; -0.2 -0.415 -0.575; -
0.1 -0.3 -0.4; -0.2 -0.415 -0.575; -0.2 -0.4 -
0.575; -0.1 -0.285 -0.415; -0.2 -0.4 -0.575; -
0.2 -0.375 -0.575; -0.1 -0.265 -0.415; -0.2 -
0.375 -0.575; -0.2 -0.35 -0.575; -0.1 -0.245 -
0.425; -0.2 -0.35 -0.575; -0.2 -0.325 -0.575; -
0.1 -0.225 -0.425; -0.2 -0.325 -0.575; -0.2 -
0.3 -0.575; -0.1 -0.21 -0.425; -0.2 -0.3 -0.575;
-0.2 -0.2 -0.445; -0.2 -0.3 -0.575; -0.25 -0.185
-0.43; -0.2 -0.3 -0.575; -0.275 -0.15 -0.435;
-0.2 -0.3 -0.575; -0.2 -0.265 -0.575; -0.275 -
0.135 -0.44;

```

Gambar A 15. Segitiga.txt (Bagian Pertama)

```
-0.2 -0.265 -0.575; -0.2 -0.1 -0.44; -0.2 -0.265
-0.575; -0.2 -0.05 -0.43; -0.2 -0.265 -0.575; -
0.2 -0.3 -0.575; -0.2 -0.05 -0.43; -0.2 -0.3 -
0.575; -0.2 -0.265 -0.575; -0.2 -0.225 -0.575;
-0.2 -0.025 -0.4275; -0.2 -0.15 -0.55; -0.2 0.0
-0.44; -0.2 -0.1 -0.55; -0.2 0.0 -0.475; -0.2 -
0.1 -0.55; -0.2 -0.075 -0.55; -0.2 0.025 -0.475;
-0.2 -0.075 -0.55; -0.2 -0.05 -0.55; -0.2 0.05
-0.485; -0.2 -0.05 -0.55; -0.2 -0.025 -0.55; -
0.2 0.075 -0.465; -0.2 0 -0.55; 0 0.15 -0.485;
-0.2 0 -0.55; -0.2 0.025 -0.55; 0 0.2 -0.5; -
0.2 0.05 -0.55; 0 0.225 -0.5;]
```

```
[-0.4 0.3 -0.5; -0.4 0.2 -0.475; -0.8 0.2 -0.4;
-0.4 0.2 -0.475; -0.4 0.185 -0.475; -0.8 0.185
-0.4; -0.4 0.15 -0.475; -0.7 0.15 -0.4; -0.4
0.15 -0.475; -0.4 0.14 -0.45; -0.7 0.14 -0.4; -
0.4 0.14 -0.45; -0.4 0.12 -0.45; -0.7 0.14 -
0.415; -0.4 0.12 -0.45; -0.4 0.1 -0.45; -0.7 0.1
-0.4; -0.4 0.1 -0.45; -0.6 0.05 -0.435; -0.75
0.05 -0.4; -0.6 0.05 -0.435; -0.6 0.025 -0.435;
-0.75 0.025 -0.41; -0.6 0.025 -0.435; -0.6 0.0
-0.435; -0.75 0.0 -0.41; -0.6 0.0 -0.435;]
```

```
[0 0 0.3; -0.2 -0.1 0.3; -0.2 0 0.285; -0.2 -
0.1 0.265; -0.3 0 0.25; -0.2 -0.1 0.235; -0.3 0
0.23; -0.2 -0.1 0.215; -0.35 0.025 0.21; -0.2 -
0.1 0.185; -0.35 0.025 0.18; -0.4 -0.1 0.17; -
0.35 0.025 0.16; -0.4 -0.1 0.15; -0.35 0.025
0.135; -0.45 -0.1 0.13; -0.5 0.025 0.12; -0.55
-0.1 0.1; -0.6 0.025 0.085; -0.65 -0.1 0.08; -
0.65 0.025 0.05; -0.65 -0.075 0.05; -0.65 0.05
0.025; -0.7 -0.05 0.01; -0.65 0.075 -0.025; -0.7 -
0.05 -0.01; -0.65 0.075 -0.025; -0.7 -0.05 -
0.03; -0.65 0.1 -0.05; -0.7 -0.025 -0.053; -0.65
0.1 -0.075; -0.7 -0.025 -0.08; -0.7 0.125 -0.1;
-0.7 -0.025 -0.09;
```

Gambar A 16. Segitiga.txt(Bagian Kedua)

```
-0.7 0.125 -0.135; -0.75 -0.015 -0.125; -0.7
0.15 -0.175; -0.75 0.0 -0.165; -0.6 0.2 -0.225;
-0.75 0.05 -0.2; -0.6 0.225 -0.265; -0.75 0.05
-0.225; -0.65 0.225 -0.3; -0.75 0.075 -0.265; -
0.675 0.25 -0.35; -0.75 0.1 -0.3; -0.675 0.275
-0.4; -0.75 0.1 -0.35; -0.675 0.275 -0.45; -0.75
0.1 -0.4;]
```

```
[0 -0.525 0.05; 0 -0.5 0.2; 0 -0.525 0.05; 0 -
0.53 0.0; 0 -0.535 0.175; 0 -0.53 0.0; -0.25 -
0.6 -0.05; 0 -0.55 0.15; -0.1 -0.6 -0.1; -0.25
-0.65 0.1; -0.1 -0.6 -0.1; -0.15 -0.625 -0.15;
-0.25 -0.67 0.1; -0.15 -0.625 -0.15; -0.15 -0.65
-0.2; -0.25 -0.7 0.05; -0.15 -0.65 -0.2; -0.15
-0.675 -0.25; -0.25 -0.725 0; -0.15 -0.7 -0.3;
-0.25 -0.75 0; -0.15 -0.7 -0.3; 0.0 -0.7 -0.275;
-0.25 -0.7 0.05; 0.0 -0.7 -0.275; 0.05 -0.7 -
0.275; -0.25 -0.725 0; 0.05 -0.7 -0.275; 0.25 -
0.7 -0.275; -0.25 -0.75 0; 0.25 -0.7 -0.275;
0.35 -0.725 -0.275; 0.0 -0.75 -0.1; 0.35 -0.725
-0.275; 0.45 -0.75 -0.275; 0.0 -0.75 -0.1;]
```

```
[-1 -1 -1; -0.4 -1 -0.8; -0.8 -1 -1.025; -0.3 -
1 -0.825; -0.6 -1 -1.05; -0.2 -1 -0.85; -0.4 -1
-1.075; -0.15 -1 -0.875; -0.2 -1 -1.1; 0.1 -1 -
0.9; 0 -1 -1.1; 0.25 -1 -0.9; 0.2 -1 -1.1; 0.35
-1 -0.9; 0.4 -1 -1.1; 0.45 -1 -0.9; 0.6 -1 -1.1;
0.55 -1 -0.9; 0.7 -1 -1.1; 0.6 -1 -0.9; 0.8 -1
-1.1; 0.75 -1 -0.9; 1 -1 -1.1; 0.9 -1 -0.9; 1.15
-1 -1.1; 1.05 -1 -0.9; 1.3 -1 -1.1; 1.2 -1 -0.9;
1.45 -1 -1.1; 1.35 -1 -0.9; 1.6 -1 -1.1; 1.5 -1
-0.9; 1.75 -1 -1.1; 1.65 -1 -0.9; 1.9 -1 -1.1;
1.8 -1 -0.9; 2.05 -1 -1.1; 1.95 -1 -0.9; 2.2 -1
-1.1; 2.1 -1 -0.9; 2.35 -1 -1.1; 2.25 -1 -0.9;
2.35 -1 -1.1; 2.4 -1 -0.9; 2.35 -1 -1.1; 2.55 -
1 -0.9; 2.35 -1 -1.1; 2.7 -1 -0.9; 2.35 -1 -1.1;
2.35 -1 -1.1; 1.2 -1 -1.1; 2.35 -1 -1.1; 1.2 -1
-1.12; 2.35 -1 -1.1; 1.2 -1 -1.14;
```

Gambar A 17. Segitiga.txt (Bagian Ketiga)

```

2.35 -1 -1.12; 1.2 -1 -1.16; 2.35 -1 -1.14; 1.2
-1 -1.18; 2.325 -1 -1.16; 0.975 -1 -1.2; 2.3 -1
-1.18; 0.95 -1 -1.22; 2.3 -1 -1.18; 0.95 -1 -
1.23; 2.275 -1 -1.18; 0.925 -1 -1.24; 2.25 -1 -
1.18; 0.9 -1 -1.26; 2.225 -1 -1.2; 0.9 -1 -1.28;
2.2 -1 -1.22; 0.875 -1 -1.3; 2.175 -1 -1.24;
0.85 -1 -1.32; 2.15 -1 -1.26; 0.875 -1 -1.34;
2.125 -1 -1.28; 0.8 -1 -1.36; 2.1 -1 -1.3; 0.75
-1 -1.38; 2.075 -1 -1.32; 0.7 -1 -1.4; 1.975 -1
-1.34; 0.675 -1 -1.42; 1.95 -1 -1.34; 0.7 -1 -
1.44; 1.925 -1 -1.34; 0.715 -1 -1.46; 1.85 -1 -
1.36; 0.715 -1 -1.48; 1.8 -1 -1.36; 0.725 -1 -
1.5; 1.775 -1 -1.38; 0.8 -1 -1.52; 1.725 -1 -
1.4; 0.75 -1 -1.54; 1.7 -1 -1.42; 0.725 -1 -
1.56; 1.675 -1 -1.44; 0.775 -1 -1.58; 1.675 -1 -
1.44; 0.8 -1 -1.6; 1.675 -1 -1.46; 0.825 -1 -
1.62; ]

```

```

[-1 -0.8 -1; -0.5 -0.5 -1; -0.775 -0.775 -1.015;
-0.5 -0.5 -1; -0.75 -0.75 -1.015; -0.5 -0.5 -1;
-0.725 -0.725 -1.03; -0.5 -0.5 -1; -0.7 -0.7 -
1.045; -0.5 -0.475 -1.015; -0.675 -0.675 -1.06;
-0.5 -0.45 -1.03; -0.65 -0.65 -1.085; -0.375 -
0.375 -1.06; -0.6 -0.6 -1.1; -0.315 -0.315 -
1.075; -0.55 -0.55 -1.115; -0.275 -0.275 -1.09;
-0.5 -0.5 -1.13; -0.15 -0.15 -1.105; -0.45 -
0.45 -1.145; -0.075 -0.075 -1.12; -0.4 -0.4 -
1.16; -0.025 -0.025 -1.135; -0.35 -0.35 -1.175;
0.025 0.025 -1.15; -0.3 -0.3 -1.19; 0.115 0.115
-1.165; -0.25 -0.25 -1.215; 0.185 0.185 -1.18;
-0.2 -0.2 -1.23; 0.225 0.25 -1.195; -0.15 -0.15
-1.245; 0.315 0.315 -1.205; -0.1 -0.1 -1.26;
0.375 0.375 -1.22; -0.1 -0.1 -1.26; 0.435 0.435
-1.235; -0.05 -0.05 -1.275; 0.49 0.49 -1.25; -
0.05 -0.05 -1.29; 0.53 0.53 -1.265; 0.0 0 -
1.315; 0.585 0.585 -1.28; 0 0 -1.33; 0.635 0.635
-1.295; 0.05 0.05 -1.345; 0.705 0.705 -1.315;

```

Gambar A 18. Segitiga.txt (Bagian Keempat)

```
0.1 0.10 -1.36; 0.78 0.78 -1.33; 0.145 0.145 -
1.375; 0.825 0.825 -1.345; 0.2 0.2 -1.39; 0.87
0.87 -1.36; 0.25 0.25 -1.415; 0.915 0.915 -
1.375; 0.3 0.3 -1.43; 0.96 0.96 -1.39; 0.35 0.35
-1.445; ]
```

```
[ 0.35 0.35 -1.445; 0.25 0.35 -1.4; 0.45 0.35 -
1.445; 0.35 0.35 -1.4; 0.55 0.35 -1.445; 0.45
0.35 -1.4; 0.65 0.35 -1.445; 0.55 0.35 -1.4;
0.65 0.3 -1.445; 0.55 0.375 -1.325; 0.65 0.25 -
1.445; 0.55 0.35 -1.325; 0.65 0.225 -1.445; 0.55
0.325 -1.325; 0.65 0.225 -1.445; 0.65 0.3 -
1.325; 0.65 0.225 -1.445; 0.675 0.275 -1.325;
0.65 0.2 -1.475; 0.675 0.25 -1.35; 0.65 0.175 -
1.475; 0.675 0.225 -1.35; 0.675 0.15 -1.475; 0.7
0.2 -1.35; 0.7 0.125 -1.475; 0.725 0.175 -1.35;
0.725 0.1 -1.475; 0.75 0.15 -1.35; 0.75 0.075 -
1.475; 0.775 0.125 -1.35; 0.775 0.05 -1.475; 0.8
0.1 -1.35; 0.8 0.025 -1.475; 0.825 0.075 -1.35;
0.825 0.0 -1.475; 0.85 0.05 -1.35; 0.85 -0.025
-1.475; 0.875 0.025 -1.35; 0.875 -0.05 -1.5; 0.9
0.0 -1.375; 0.9 -0.075 -1.5; 0.925 -0.025 -
1.375; 0.925 -0.125 -1.525; 0.95 -0.075 -1.375;
0.95 -0.15 -1.525; 0.975 -0.1 -1.375; 0.975 -
0.175 -1.525; 1 -0.125 -1.375; 1 -0.2 -1.525;
1.05 -0.15 -1.375; 1.05 -0.225 -1.55; 1.1 -0.175
-1.4; 1.1 -0.25 -1.55; 1.15 -0.2 -1.4; 1.125 -
0.275 -1.55; 1.175 -0.225 -1.4; 1.15 -0.3 -1.55;
1.2 -0.25 -1.4; 1.175 -0.325 -1.55; 1.225 -0.275
-1.4; 1.2 -0.35 -1.55; 1.25 -0.3 -1.4; 1.225 -
0.375 -1.55; 1.275 -0.325 -1.4; 1.25 -0.4 -1.55;
1.3 -0.35 -1.4; 1.275 -0.425 -1.575; 1.325 -
0.375 -1.425; 1.3 -0.45 -1.575; 1.35 -0.4 -
1.425; 1.325 -0.475 -1.575; 1.375 -0.425 -1.425;
1.35 -0.5 -1.575; 1.4 -0.45 -1.425; 1.375 -0.525
-1.575;
```

Gambar A 19. Segitiga.txt (Bagian Kelima)

```

1.425 -0.475 -1.425; 1.4 -0.55 -1.575; 1.45 -
0.5 -1.425; 1.425 -0.575 -1.575; 1.475 -0.525 -
1.425; 1.45 -0.6 -1.575; 1.5 -0.55 -1.425; 1.475
-0.625 -1.6; 1.525 -0.575 -1.45; 1.5 -0.65 -1.6;
1.55 -0.6 -1.45; 1.525 -0.675 -1.6; 1.575 -0.625
-1.45; 1.55 -0.7 -1.6; 1.6 -0.65 -1.45; 1.575 -
0.725 -1.6; 1.625 -0.675 -1.45; 1.575 -0.75 -
1.575; 1.625 -0.675 -1.45; 1.575 -0.775 -1.55;
1.625 -0.675 -1.45; 1.575 -0.8 -1.525; 1.625 -
0.675 -1.45; 1.575 -0.825 -1.5; 1.625 -0.675 -
1.45; 1.6 -0.85 -1.5; 1.65 -0.7 -1.45;]

[ -0.15 -0.15 -1.245; -0.15 -0.075 -1.2; -0.025
-0.15 -1.245; -0.025 -0.075 -1.2; 0.1 -0.15 -
1.245; 0.1 -0.075 -1.2; 0.225 -0.15 -1.245;
0.225 -0.075 -1.2; 0.25 -0.175 -1.245; 0.25 -
0.1 -1.2; 0.275 -0.2 -1.245; 0.275 -0.125 -1.2;
0.3 -0.225 -1.245; 0.3 -0.15 -1.2; 0.325 -0.25
-1.245; 0.325 -0.175 -1.2; 0.35 -0.275 -1.245;
0.35 -0.2 -1.2; 0.375 -0.3 -1.245; 0.375 -0.225
-1.2; 0.4 -0.325 -1.245; 0.4 -0.25 -1.2; 0.425
-0.35 -1.245; 0.425 -0.275 -1.2; 0.45 -0.375 -
1.245; 0.45 -0.3 -1.2; 0.475 -0.4 -1.245; 0.475
-0.325 -1.2; 0.5 -0.425 -1.245; 0.5 -0.35 -1.2;
0.525 -0.45 -1.245; 0.525 -0.375 -1.2; 0.575 -
0.475 -1.245; 0.575 -0.4 -1.2; 0.6 -0.5 -1.245;
0.6 -0.425 -1.2; 0.625 -0.525 -1.245; 0.625 -
0.45 -1.2; 0.65 -0.55 -1.245; 0.65 -0.475 -1.2;
0.675 -0.575 -1.245; 0.675 -0.5 -1.2; 0.7 -0.6
-1.245; 0.7 -0.525 -1.2; 0.725 -0.625 -1.245;
0.725 -0.55 -1.2; 0.75 -0.65 -1.245; 0.75 -0.575
-1.2; 0.775 -0.675 -1.245; 0.775 -0.6 -1.2; 0.8
-0.7 -1.245; 0.8 -0.625 -1.2; 0.8 -0.725 -1.245;
0.8 -0.65 -1.2; 0.8 -0.75 -1.245; 0.8 -0.675 -
1.2; 0.8 -0.775 -1.245; 0.8 -0.7 -1.2; 0.8 -0.8
-1.245;

```

Gambar A 20. Segitiga.txt (Bagian Keenam)


```
0.8 -0.725 -1.2; 0.8 -0.825 -1.245; 0.8 -0.75 -
1.2; 0.825 -0.85 -1.245; 0.825 -0.775 -1.2; 0.85
-0.875 -1.245; 0.85 -0.8 -1.2; 0.875 -0.9 -
1.275; 0.875 -0.825 -1.175; 0.9 -0.925 -1.275;
0.9 -0.85 -1.175; 0.925 -0.95 -1.275; 0.925 -
0.875 -1.175; 0.95 -0.975 -1.275; 0.95 -0.9 -
1.175; 0.975 -1 -1.275; 0.975 -0.925 -1.175; 1
-1.025 -1.275; 1 -0.95 -1.175; 1.025 -1.05 -
1.275; 1.025 -0.975 -1.175; 1.05 -1.075 -1.275;
1.05 -1 -1.175; ]
```

Gambar A 21. Segitiga.txt (Bagian Ketujuh)

(halaman ini sengaja dikosongkan)

LAMPIRAN B. MATLAB Function

1	function fcn(C0,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11, C12,C13,C14,C15,C16,C17,C18,C19,C20,C21,C2 2,C23,C24,C25,C26,C27,C28,C29,C30,C31,C32)
2	hold on;
3	if(size(C6,1)>1)
4	plot3(C6(:,1),C6(:,2),C6(:,3),'Color',C5,' LineWidth',C4);
5	Else
6	plot3(C6(1),C6(2),C6(3),'Color',C5,'LineWi dth',C4);
7	end;
8	if(size(C7,1)>1)
9	plot3(C7(:,1),C7(:,2),C7(:,3),'Color',C5,' LineWidth',C4);
10	else
11	plot3(C7(1),C7(2),C7(3),'Color',C5,'LineWi dth',C4);
12	end;
13	if(size(C8,1)>1)
14	plot3(C8(:,1),C8(:,2),C8(:,3),'Color',C5,' LineWidth',C4);
15	else
16	plot3(C8(1),C8(2),C8(3),'Color',C5,'LineWi dth',C4);
17	end;
18	if(size(C9,1)>1)
19	plot3(C9(:,1),C9(:,2),C9(:,3),'Color',C5,' LineWidth',C4);
20	else
21	plot3(C9(1),C9(2),C9(3),'Color',C5,'LineWi dth',C4);
22	end;

Kode Sumber B 1. MATLAB Function (Bagian Pertama)

23	if(size(C10,1)>1)
24	plot3(C10(:,1),C10(:,2),C10(:,3),'Color',C5,'LineWidth',C4);
25	else
26	plot3(C10(1),C10(2),C10(3),'Color',C5,'LineWidth',C4);
27	end;
28	if(size(C11,1)>1)
29	plot3(C11(:,1),C11(:,2),C11(:,3),'Color',C5,'LineWidth',C4);
30	else
31	plot3(C11(1),C11(2),C11(3),'Color',C5,'LineWidth',C4);
32	end;
33	if(size(C12,1)>1)
34	plot3(C12(:,1),C12(:,2),C12(:,3),'Color',C5,'LineWidth',C4);
35	else
36	plot3(C12(1),C12(2),C12(3),'Color',C5,'LineWidth',C4);
37	end;
38	if(size(C13,1)>1)
39	plot3(C13(:,1),C13(:,2),C13(:,3),'Color',C5,'LineWidth',C4);
40	else
41	plot3(C13(1),C13(2),C13(3),'Color',C5,'LineWidth',C4);
42	end;
43	if(size(C14,1)>1)
44	plot3(C14(:,1),C14(:,2),C14(:,3),'Color',C5,'LineWidth',C4);
45	else
46	plot3(C14(1),C14(2),C14(3),'Color',C5,'LineWidth',C4);
47	end;

Kode Sumber B 2. MATLAB Function (Bagian Kedua)

48	if(size(C15,1)>1)
49	plot3(C15(:,1),C15(:,2),C15(:,3),'Color',C5,'LineWidth',C4);
50	else
51	plot3(C15(1),C15(2),C15(3),'Color',C5,'LineWidth',C4);
52	end;
53	if(size(C16,1)>1)
54	plot3(C16(:,1),C16(:,2),C16(:,3),'Color',C5,'LineWidth',C4);
55	else
56	plot3(C16(1),C16(2),C16(3),'Color',C5,'LineWidth',C4);
57	end;
58	if(size(C17,1)>1)
59	plot3(C17(:,1),C17(:,2),C17(:,3),'Color',C5,'LineWidth',C4);
60	else
61	plot3(C17(1),C17(2),C17(3),'Color',C5,'LineWidth',C4);
62	end;
63	if(size(C18,1)>1)
64	plot3(C18(:,1),C18(:,2),C18(:,3),'Color',C5,'LineWidth',C4);
65	else
66	plot3(C19(1),C19(2),C19(3),'Color',C5,'LineWidth',C4);
67	end;
68	if(size(C20,1)>1)
69	plot3(C20(:,1),C20(:,2),C20(:,3),'Color',C5,'LineWidth',C4);
70	else
71	plot3(C20(1),C20(2),C20(3),'Color',C5,'LineWidth',C4);
72	end;

Kode Sumber B 3. MATLAB Function (Bagian Ketiga)

73	if(size(C21,1)>1)
74	plot3(C21(:,1),C21(:,2),C21(:,3),'Color',C5,'LineWidth',C4);
75	else
76	plot3(C21(1),C21(2),C21(3),'Color',C5,'LineWidth',C4);
77	end;
78	if(size(C22,1)>1)
79	plot3(C22(:,1),C22(:,2),C22(:,3),'Color',C5,'LineWidth',C4);
80	else
81	plot3(C22(1),C22(2),C22(3),'Color',C5,'LineWidth',C4);
82	end;
83	if(size(C23,1)>1)
84	plot3(C23(:,1),C23(:,2),C23(:,3),'Color',C5,'LineWidth',C4);
85	else
86	plot3(C23(1),C23(2),C23(3),'Color',C5,'LineWidth',C4);
87	end;
88	if(size(C24,1)>1)
89	plot3(C24(:,1),C24(:,2),C24(:,3),'Color',C5,'LineWidth',C4);
90	else
91	plot3(C24(1),C24(2),C24(3),'Color',C5,'LineWidth',C4);
92	end;
93	if(size(C25,1)>1)
94	plot3(C25(:,1),C25(:,2),C25(:,3),'Color',C5,'LineWidth',C4);
95	else
96	plot3(C25(1),C25(2),C25(3),'Color',C5,'LineWidth',C4);
97	end;
98	if(size(C26,1)>1)

Kode Sumber B 4. MATLAB Function (Bagian Keempat)

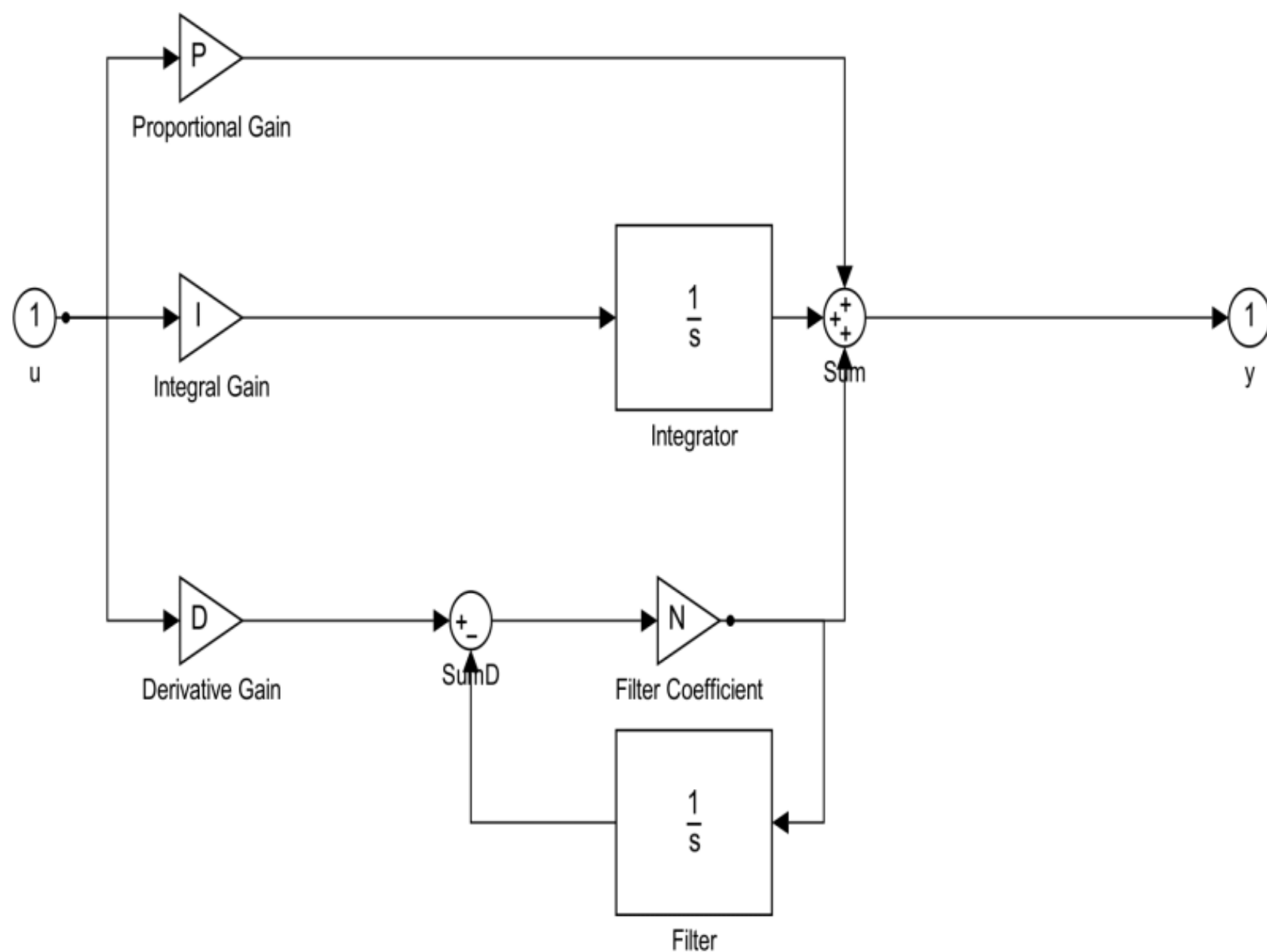
99	plot3(C26(:,1),C26(:,2),C26(:,3),'Color',C5,'LineWidth',C4);
100	else
101	plot3(C26(1),C26(2),C26(3),'Color',C5,'LineWidth',C4);
102	end;
103	if(size(C27,1)>1)
104	plot3(C27(:,1),C27(:,2),C27(:,3),'Color',C5,'LineWidth',C4);
105	else
106	plot3(C27(1),C27(2),C27(3),'Color',C5,'LineWidth',C4);
107	end;
108	if(size(C28,1)>1)
109	plot3(C28(:,1),C28(:,2),C28(:,3),'Color',C5,'LineWidth',C4);
110	else
111	plot3(C28(1),C28(2),C28(3),'Color',C5,'LineWidth',C4);
112	end;
113	if(size(C29,1)>1)
114	plot3(C29(:,1),C29(:,2),C29(:,3),'Color',C5,'LineWidth',C4);
115	else
116	plot3(C29(1),C29(2),C29(3),'Color',C5,'LineWidth',C4);
117	end;
118	if(size(C30,1)>1)
119	plot3(C30(:,1),C30(:,2),C30(:,3),'Color',C5,'LineWidth',C4);
120	else
121	plot3(C30(1),C30(2),C30(3),'Color',C5,'LineWidth',C4);
122	end;
123	if(size(C31,1)>1)

Kode Sumber B 5. MATLAB Function (Bagian Kelima)

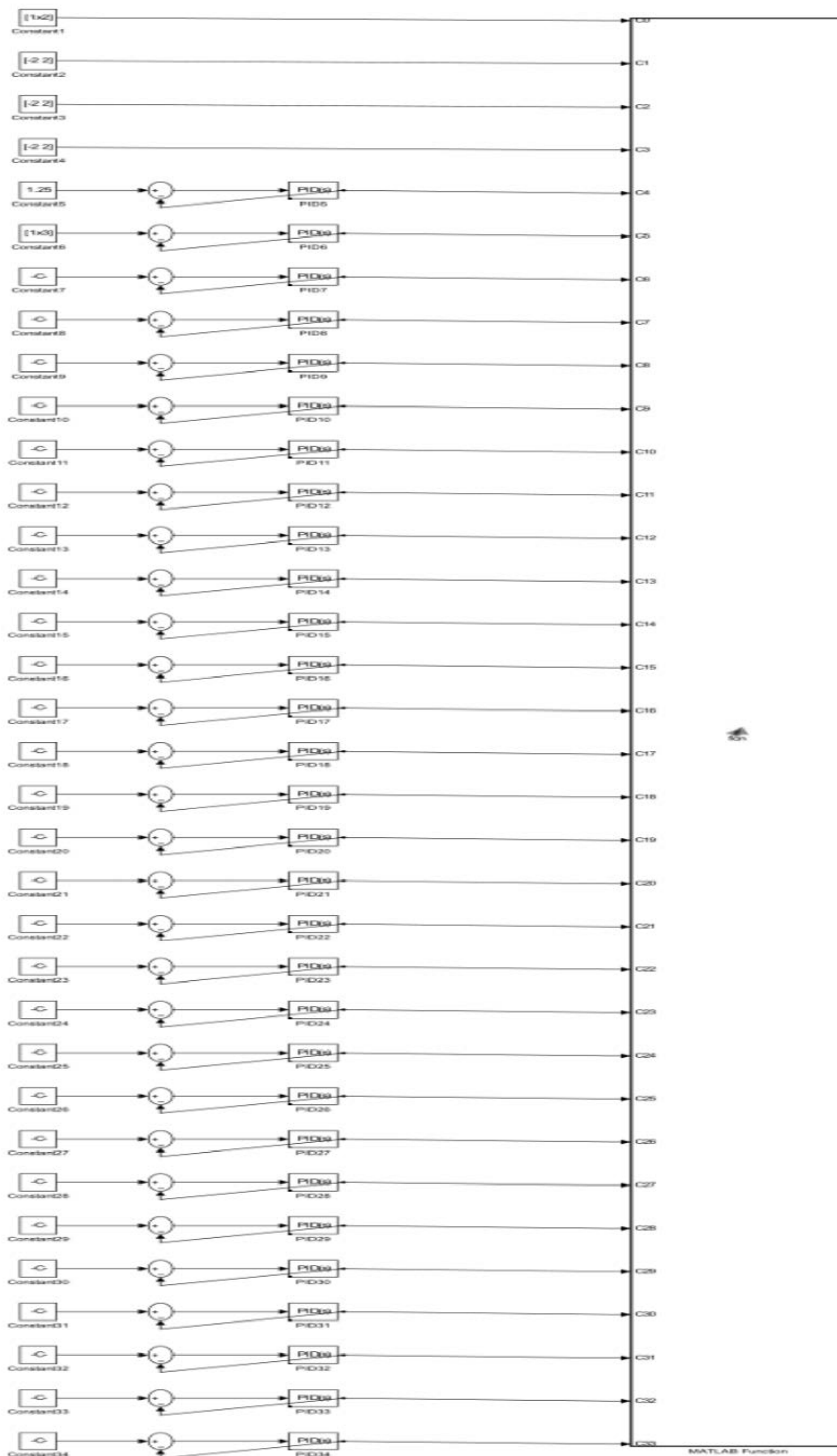
124	plot3(C31(:,1),C31(:,2),C31(:,3),'Color',C5,'LineWidth',C4);
125	else
126	plot3(C31(1),C31(2),C31(3),'Color',C5,'LineWidth',C4);
127	end;
128	if(size(C32,1)>1)
129	plot3(C32(:,1),C32(:,2),C32(:,3),'Color',C5,'LineWidth',C4);
130	else
131	plot3(C32(1),C32(2),C32(3),'Color',C5,'LineWidth',C4);
132	end;
133	if(size(C33,1)>1)
134	plot3(C33(:,1),C33(:,2),C33(:,3),'Color',C5,'LineWidth',C4);
135	else
136	plot3(C33(1),C33(2),C33(3),'Color',C5,'LineWidth',C4);
137	end;
138	set(gca,'YTicklabel',[]);
139	set(gca,'YTick',[]);
140	set(gca,'XTicklabel',[]);
141	set(gca,'XTick',[]);
142	set(gca,'ZTicklabel',[]);
143	set(gca,'ZTick',[]);
144	xlim(C1);
145	ylim(C2);
146	zlim(C3);
147	az=C0(1);
148	el = C0(2);
149	view(az,el);

Kode Sumber B 6. MATLAB Function (Bagian Keenam)

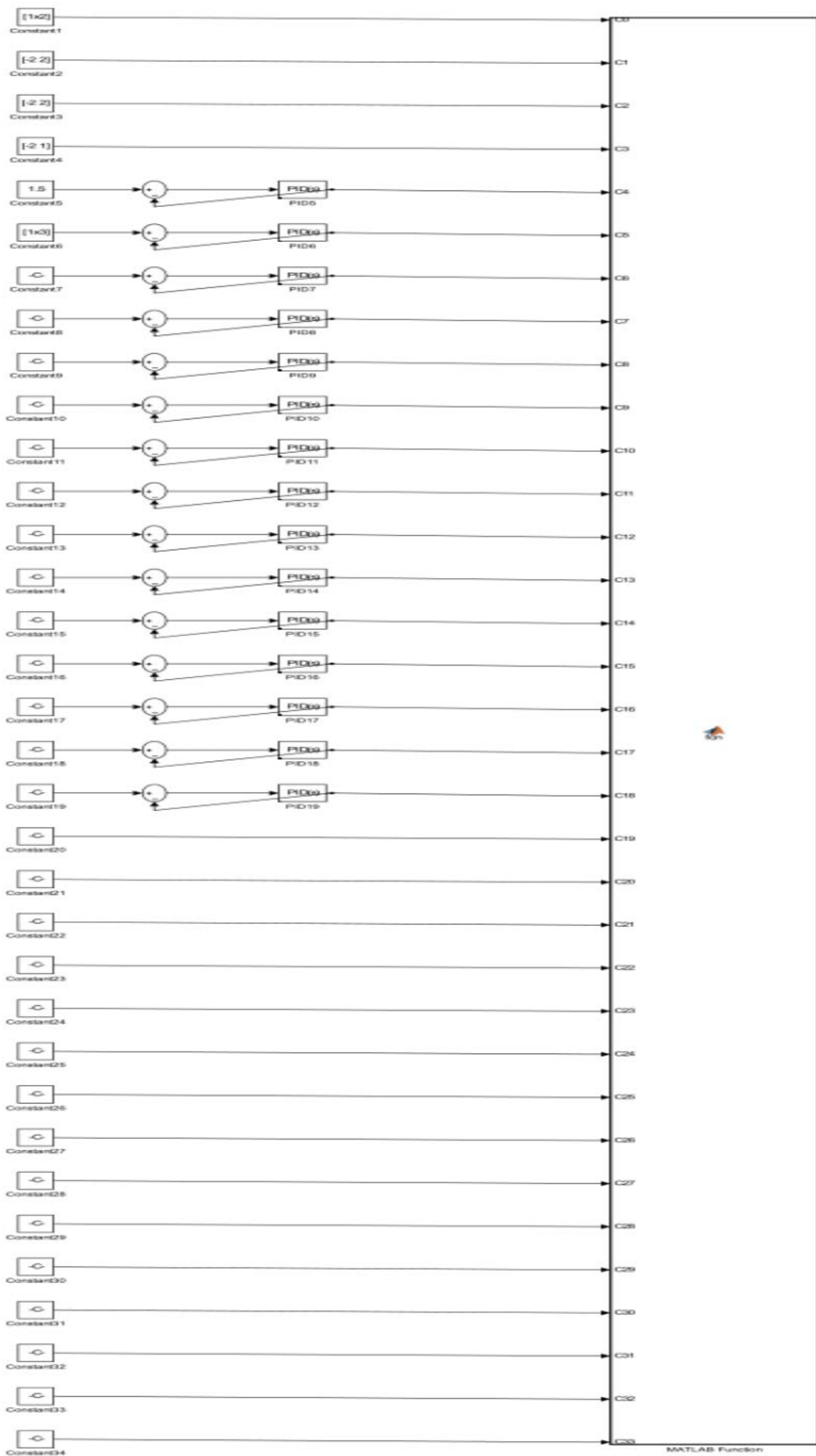
Lampiran C. Implementasi Sistem



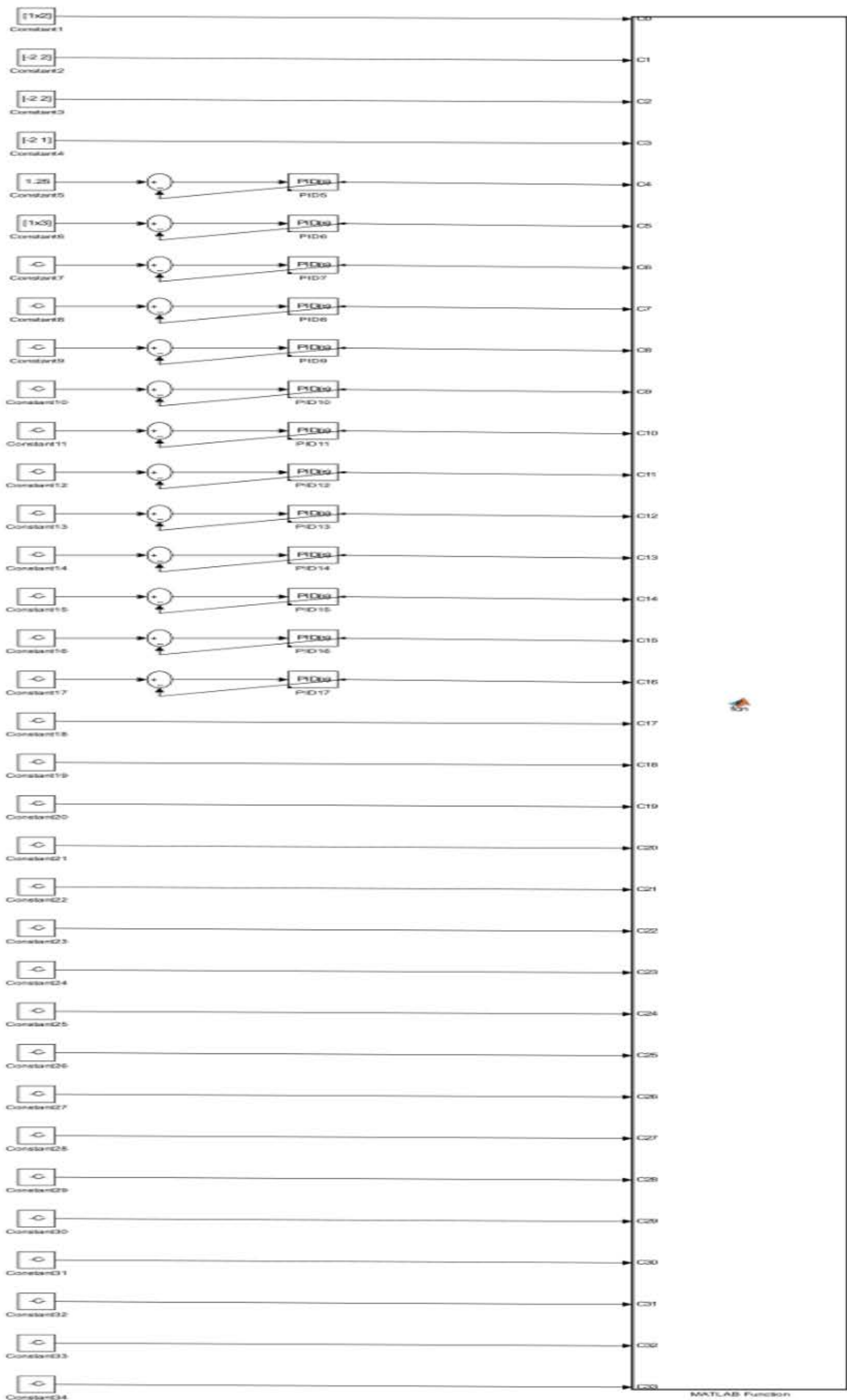
Gambar C 1. Implementasi Sistem Block Diagram PID



86 Gambar C 2. Implementasi Sistem Dengan Input Cube.txt



Gambar C 3. Implementasi Sistem Dengan Input Segilima.txt



BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Dalam proses pengerjaan tugas akhir mulai dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Untuk mengimplementasikan pena dinamis dibutuhkan dua komponen utama, yaitu *muscle model* dan pergerakan pena. *Muscle model* dimodelkan dan diimplementasikan pada PID, sedangkan pergerakan pena diimplementasikan oleh ode *solver*.
2. Pada penerapan *muscle model*, parameter PID yang lebih baik adalah parameter PID yang telah di-*tuning*.
3. Pada penggunaan *solver* untuk mensimulasikan gerakan pena, Ode113 dan Ode23 memiliki performa lebih baik daripada Ode45.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.

1. Untuk penggunaan *solver* yang lebih baik, menggunakan MATLAB versi terbaru karena ada *auto-solver*.
2. Untuk performa yang lebih baik, dapat menggunakan *library* dasar dari SIMULINK.

(halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] D. H. House dan M. Singh, "Line drawing as a dynamic process," *Pacific Graphics*, vol. 07, p. 351–360, 2007.
- [2] G. Orbay dan L. Burak Kara, "Pencil-like sketch rendering of 3D scenes using trajectory planning and dynamic tracking," *Journal of Visual Languages and Computing*, vol. 25, pp. 488-489, 2014.
- [3] M. Zahim Sujod, "EGR 326, Fall 2013," September 2013. [Online]. Available: <http://www.science.smith.edu/~jcardell/Courses/EGR326/MatlabSimulinkTutorial.ppt>. [Diakses 8 Juni 2016].
- [4] M. Inc., "MathWorks Documentation: Plot3," [Online]. Available: <http://www.mathworks.com/help/matlab/ref/plot3.html>. [Diakses 8 Juni 2016].
- [5] M. Inc., "MathWorks Documentation: Choose a Solver," [Online]. Available: <http://www.mathworks.com/help/simulink/ug/types-of-solvers.html>. [Diakses 20 April 2016].
- [6] M. Inc., "MathWorks Documentation: Exploring Variable-Step Solvers Using a Stiff Model," [Online]. Available: <http://www.mathworks.com/help/simulink/examples/exploring-variable-step-solvers-using-a-stiff-model.html>. [Diakses 9 Juni 2016].
- [7] G. Wong dan J. Wang, "Model-Based Control," dalam *Real-Time Rendering: Computer Graphics with Control Engineering*, CRC Press, 2013, pp. 68-83.
- [8] J. Hogenson, "PID for Dummies," [Online]. Available: http://www.csimn.com/CSI_pages/PIDforDummies.html. [Diakses 6 6 2016].

- [9] K. H. Ang, G. Chong dan Y. Li, "PID Control System Analysis, Design, and Technology," *IEEE Transactions On Control Systems Technology*, vol. 13, p. 560, 2005.
- [10] "Ziegler Nichols method," [Online]. Available: http://kurser.iha.dk/m/mtpri1/control/3_PID/f_ziegler/ziegler_nichols_method.html. [Diakses 6 6 2016].
- [11] A. Hill, "The Heat of Shortening and the Dynamic Constants of Muscle," *Proceedings of the Royal Society B*, pp. 136-195, 1938.

BIODATA PENULIS



Penulis dilahirkan di Surabaya, 9 Februari 1994, merupakan anak keempat dari 4 bersaudara. Penulis telah menempuh pendidikan formal yaitu TK Permata Bunda(1998-2000), SDK St. Theresia II Surabaya (2000-2006), SMPK St. Stanislaus II Surabaya (2006-2009), SMAK St. Stanislaus Surabaya (2009-2012), dan mahasiswa S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya rumpun mata kuliah

Interaksi, Grafika, dan Seni (2012-2016).

Penulis dapat dihubungi melalui *surel* klabrahams444@gmail.com.